Teresa M. Corbin (SBN 132360)
Christopher Kelley (SBN 166608)
Thomas C. Mavrakakis (SBN 177927)
HOWREY SIMON ARNOLD & WHITE, LLP
301 Ravenswood Avenue
Menlo Park, California 94025
Telephone: (650) 463-8100
Facsimile: (650) 463-8400

Attorneys for Defendants
AEROFLEX, INC.,
AMI SEMICONDUCTOR, INC.,
MATROX ELECTRONIC SYSTEMS LTD.,
MATROX GRAPHICS INC.
MATROX INTERNATIONAL CORP., and
MATROX TECH, INC.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

RICOH COMPANY, LTD., Plaintiff, vs. AEROFLEX INCORPORATED, AMI SEMICONDUCTOR, INC., MATROX ELECTRONIC SYSTEMS, LTD., MATROX GRAPHICS INC., MATROX INTERNATIONAL CORP., and MATROX TECH, INC., Defendants.	Case No. C03-4669 MJJ **DEFENDANTS' NOTICE OF MOTION AND MOTION FOR JUDGMENT ON THE PLEADINGS PURSUANT TO RULE 12(C)** Date: March 16, 2004 Time: 9:30 a.m. Ctrm: 11, 19th Floor Judge: Hon. Martin J. Jenkins

TABLE OF CONTENTS

TABLE OF AUTHORITIES

CASES

STATUTES

RULES

NOTICE OF MOTION AND RELIEF SOUGHT

Please take notice that, on March 16, 2004 at 9:30 a.m., before the Honorable Martin J. Jenkins in Courtroom 11, 19th Floor, in the United States District Court, 450 Golden Gate Avenue, San Francisco, California, Defendants Aeroflex, Inc., AMI Semiconductor, Inc., Matrox Electronic Systems, Ltd., Matrox Graphics Inc., Matrox International Corp., and Matrox Tech, Inc. (the "ASIC Defendants") will each seek a judgment from the Court, pursuant to Rule 12(c) of the Federal Rules of Civil Procedure, that they do not infringe claims 13-20 of United States Patent No. 4,922,432 (the "'432 patent") under 35 U.S.C. § 271(g).

This motion is based upon the following memorandum of points and authorities, the complaint (Attachment A to the present motion) in this action which includes a copy of Ricoh's '432 patent (Exhibit 1 to Attachment A), the oral arguments of counsel at the hearing on this motion, and all other pleadings in this action.

MEMORANDUM OF POINTS AND AUTHORITIES IN SUPPORT OF MOTION

I. STATEMENT OF THE LEGAL ISSUE TO BE DECIDED

Whether alleging that a product is "designed by or using information generated by" a process patented in the United States fails to state a claim upon which relief may be granted under 35 U.S.C. 271(g), which requires "a product which is made by a process patented in the United States?"

Last summer, the Federal Circuit, affirming a dismissal under Rule 12(b)(6) of a patent infringement claim arising under 35 U.S.C. § 271(g), confirmed that "made by" under that section required that "the process must be used directly in the manufacture of the product, and not merely as a predicate process to identify the product to be manufactured." *Bayer AG v. Housey Pharmaceuticals, Inc.*, 340 F. 3d 1367, 1378 (Fed. Cir. 2003). Here Ricoh's infringement allegations are limited to Section 271(g) because Ricoh only alleges that the ASIC Defendants are using, offering to sell, selling, and/or importing into the United States products (i.e., ASICs) "designed by or using information generated by" Ricoh's patented processes. Ricoh's complaint, therefore, is legally insufficient since it merely alleges that the ASICs are "designed by" the patented processes not "made by" them as Section 271(g) requires. Ricoh also cannot amend its complaint to properly allege that the ASICs are "made

by" the processes of claims 13-20 of the '432 patent because those claims are directed to processes for designing ASICs, which are predicate processes that are not steps in making the ASICs.

Given the legal insufficiency of Ricoh's infringement allegations under Section 271(g) and Ricoh's inability to amend them to state a claim under that section, should the Court enter judgment for the ASIC Defendants on Ricoh's patent infringement complaint pursuant to Rule 12(c) of the Federal Rules of Civil Procedure?

II. INTRODUCTION

As pled, Ricoh's patent infringement claims, made solely under 35 U.S.C. § 271(g), simply cannot be pursued as a matter of law and should be dismissed pursuant to Rule 12(c). Specifically, the Federal Circuit's recent decision in *Bayer AG v. Housey Pharmaceuticals, Inc.* construes Section 271(g) to be limited to patented processes for making products. The '432 patent's claimed processes are, however, for designing ASICs only. Therefore, the only infringement allegations in Ricoh's complaint are not legally cognizable, and much of the burdensome discovery that Ricoh seeks is legally irrelevant.

The ASIC Defendants bring this motion now because this relatively simple legal dispute between the parties regarding the scope of 35 U.S.C § 271(g) will have a profound impact on what is at issue in this action. Specifically, the ASIC Defendants are mindful that the Court may, in its discretion, decide to allow Ricoh to amend its complaint to add infringement allegations pursuant to Section 271(a) for at least some of the ASIC Defendants. An infringement claim under Section 271(a), however, would be necessarily limited to the ASIC Defendants who actually use the '432's patented processes in the United States. The scope of such an action under Section 271(a) would be much narrower than the current action pled under Section 271(g) by Ricoh. Regardless of whether the Court allows Ricoh to amend to add claims pursuant to Section 271(a), however, the present action as pled by Ricoh under Section 271(g) should be dismissed.

III. SUMMARY OF ARGUMENT

Ricoh's patented process is for designing application specific integrated circuits (i.e., ASICs). In particular, claims 13-20 of Ricoh's '432 patent are directed to processes for designing ASICs and not for manufacturing them. These processes produce design information or data that at best can be

used to manufacture masks—not ASICs. These masks are in turn used by the manufacturing processes, which make the ASICs. Because the patented processes of the '432 patent are for designing ASICs and they are not steps in their actual manufacture, Ricoh, as a matter of law, cannot allege a legally sufficient claim for patent infringement under Section 271(g) for that patent.

Despite the fact that the patented processes of the '432 patent cannot support a claim under Section 271(g), Ricoh's complaint only alleges infringing acts under that section. Specifically, Ricoh's infringement claim is based on the ASIC Defendants use, sale, and/or importing of ASICs "designed by or using information generated by" software products that they license from Synopsys—e.g., Design Compiler. (Complaint at ¶¶ 16, 22, 28, 34, 40, and 46). Thus, Ricoh has intentionally and expressly chosen to limit its infringement allegations to the allegedly infringing acts of using, selling, offering to sell, and/or importing ASICs, and has completely failed to allege as an act of infringement under Section 271(a) that the ASIC Defendants have used the '432's patented process in the United States.

The underlying reason for this decision by Ricoh to rely solely on infringing acts under Section 271(g) is readily apparent and highlights the importance of this Court's resolution of the legal issue raised by the present motion. Specifically, Ricoh's damages theory, albeit flawed, seeks to recover damages based on the importing, using, and/or selling of ASICs by the ASIC Defendants in the United States. Ricoh is seeking these damages as opposed to damages based on the use of the software that allegedly performs the patented processes of the '432 patent in the United States-i.e., the licensing fees paid by the defendants to Synopsys for that software. More specifically, by alleging infringement based on Section 271(g) only against all of the ASIC Defendants, Ricoh seeks to encompass as infringing acts the use and/or sale of ASICs in the United States under Section 271(g) where the accused process is practiced both abroad as well as within the United States. This is contrary to the purpose of Section 271(g), as stated in the *Housey* decision, which was to address the concern that competitors could avoid infringement by practicing patented processes abroad and then use or sell products made by those processes in the United States. Thus, Ricoh's inability to allege infringement against the ASIC Defendants under Section 271(g), as a matter of law, will also unquestionably have a profound effect on the scope of the damages issues in the present action as well.

Unfortunately for Ricoh, the Federal Circuit has construed Section 271(g) to be limited to processes "used directly in the manufacture" of the products. Ricoh's current infringement allegations merely allege that the ASICs used, sold, offered for sale, or imported into the United States are "designed by" as opposed to "made by" its patented processes. This is insufficient as a matter of law. The ASIC Defendants are, therefore, entitled to the entry of judgment dismissing all of the infringement claims in Ricoh's complaint pursuant to Rule 12(c).

Besides the fact that Ricoh's allegations are legally insufficient as pled, the '432 patent's design processes are predicate processes for identifying the product to be manufactured and are not processes used directly in the manufacture of products. In other words, designing is a predicate process that is done once to identify and generate data for the product to be made and therefore, is not a step in the manufacture of that final product. Thus, attempting to amend Ricoh's legally insufficient claim to state an infringement claim pursuant to Section 271(g) would be futile, since Ricoh cannot allege such a claim for the design processes of its '432 patent.

IV. ARGUMENT

A. Legal Standard For Rule 12(c) Motions For Judgment On The Pleadings

"Judgment on the pleadings is proper when there are no issues of material fact, and the moving party is entitled to judgment as a matter of law." *3550 Stevens Creek Associates v. Barclays Bank of California*, 915 F.2d 1355, 1357 (9th Cir. 1990) (citing Fed. R. Civ. P. 12(c)). Here the ASIC defendants motion for judgment on the pleadings is based solely on the resolution of a question of law—i.e., whether an allegation that a product is "designed by" a patented process is legally sufficient to state an infringement claim under a statute that requires that the product be "made by" the patented process.

To resolve this issue of law the Court may rely on the patent that is the subject of the infringement claim without converting the motion on the pleadings to one for summary judgment. Specifically, documents that are properly attached as exhibits to a complaint can be considered on a Rule 12(c) motion without converting that motion to a Rule 56 motion for summary judgment. *See Qwest Comm. Corp. v. City of Berkeley, et al.*, 208 F.R.D. 288, 291 (N.D. Cal. 2002) ("materials properly attached to a complaint may be considered") (citing *Amfac Mortg. Corp. v. Arizona Mall of*

Tempe, Inc., 583 F.2d 426, 429 (9th Cir. 1978)). Furthermore, documents can be "considered if the documents' 'authenticity . . . is not contested' and the 'plaintiff's complaint necessarily relies' on them." *Lee v. City of Los Angeles, et al.*, 250 F.3d 668, 688 (9th Cir. 2001) (citations omitted).

Consequently, the Court, without converting this motion to a Rule 56 motion, may consider and rely on the '432 patent to grant the ASIC Defendants Rule 12(c) motion. The '432 patent was properly attached to Ricoh's complaint. Furthermore, the '432 patent is unquestionably central to Ricoh's patent infringement claims and Ricoh has no basis for disputing its own patent's authenticity.

B. Ricoh's Infringement Allegations Are Limited To Infringing Acts Under § 271(g)

A process patented in the United States may be infringed by: 1) using that claimed process in the United States under 35 U.S.C. § 271(a); or 2) using, offering to sell, selling within the United States, or importing into the United States "a product which is made by a process patented in the United States" under 35 U.S.C. §271(g). 35 U.S.C. § 271(a), (c). Here Ricoh's infringement allegations, which are identical for each of the ASIC Defendants in the present action, fail to allege that the ASIC Defendants use the '432 patent's claimed processes in the United States under Section 271(a), and only allege infringement under Section 271(g). (Complaint at ¶¶ 16, 22, 28, 34, 40, and 46).

The chart below, which depicts Ricoh's boiler-plate infringement allegation for each of the ASIC Defendants on the left-hand-side and the pertinent portion of 35 U.S.C. § 271(g) on the right-hand-side, demonstrates that Ricoh's infringement allegations are limited to claims under Section 271(g). Specifically, the **bolded** and <u>underlined</u> portions in that chart illustrate the correspondence between Ricoh's infringement allegation and Section 271(g).

Ricoh's Infringement Allegations	35 U.S.C. § 271(g)
Upon information and belief, [the defendant] has been and is now infringing the '432 patent by **using**, **offering to sell**, and/or by **selling** and/or **importing** into the United States <u>application specific integrated circuits designed by or using information generated by, the process of one or more of claims 13-20 of the '432 patent</u>, either literally or under the doctrine of equivalents. (Complaint at ¶¶ 16, 22, 28, 34, 40, and 46).	Whoever without authority **imports** into the United States or **offers to sell**, **sells**, or **uses** within the United States <u>a product which is made by a process patented in the United States</u> shall be liable as an infringer, if the importation, offer to sell, sale, or use of the product occurs during the term of such process patent. 35 U.S.C. § 271(g).

Furthermore, this chart illustrates that Ricoh fails to make any allegation of infringement under Section 271(a). Section 271(a) provides that "whoever without authority makes, uses, offers to sell, or sells any patented invention, within the United States or imports into the United States any patented invention during the term of the patent therefore, infringes the patent." 35 U.S.C. § 271(a). With respect to patented processes, there is direct infringement under Section 271(a) only when that process is performed in the United States. *See Joy Technologies, Inc. v. Flakt, Inc.*, 6 F.3d 770, 773 (Fed. Cir. 1993). However, as this chart illustrates, Ricoh fails to allege that the ASIC Defendants infringe the '432 patent by using the patented process in the United States.

Not only are Ricoh's infringement allegations limited to Section 271(g), but they also are limited in that they only allege the importing, use, sale, and offer to sell of ASICs—not the design information or data generated by the '432's patented processes. (Complaint at ¶¶ 16, 22, 28, 34, 40, and 46). In other words, Ricoh's Section 271(g) allegations do not allege that the information or data generated by the '432's patented processes are "manufactured products" within the meaning of that section. This is also not surprising since the Federal Circuit has confirmed that Section 271(g) is limited to manufactured products and that information or data is not a manufactured product under that section. *See Housey*, 340 F.3d at 1371.[1]

C. The '432 Patent's Process Claims 13-20 Are Processes For Generating Design Information—Not For Manufacturing ASICs

Claims 13-20 are all explicitly directed to a "design process for designing an application specific integrated circuit" (an "ASIC")—not a process for manufacturing them. ('432 patent at Col. 16, lines 34-35, 66; Col. 17, lines 1, 4, 8, 11-12; Col. 18, lines 15, 22). That Claims 13-20 are not processes for manufacturing ASICs is also demonstrated by the fact that these processes only generate design information or data that—at best—can only be used to make masks. (*See e.g.*, '432 patent, Claim 13 at Col. 16, lines 61-65; Claim 14 at Col. 16, lines 66-68) (claim 14's process generates mask data from a netlist that claim 13's process generates).

---

[1] Ricoh's complaint against the ASIC Defendants was filed months before the Federal Circuit's decision. **[Is this footnote necessary?]**

In particular, the processes of claims 13 and 15-20 only generate a netlist. ('432 patent at Col. 16, lines 61-65; Col. 17, lines 1-11; Col. 18, lines 10-24). A netlist is nothing more than a structural level design specification of an integrated circuit—i.e., a list identifying the hardware cells and their interconnection requirements. (*Id*; '432 patent, Abstract at lines 17-18) ("This list of hardware cells and their interconnection requirements is set forth in a netlist."); ('432 patent, Summary Of The Invention at Col. 2, lines 43-45) ("The list of hardware cells and their interconnection requirements may be represented in the form of a netlist."); (*See also*, '432 patent at Col. 1, lines 38-40). The process of claim 14 merely generates "mask data" from the netlist generated by the process of claim 13. ('432 patent at Col. 16, lines 66-68). According to the '432 patent, mask data is geometrical information, which provides the physical layout level description of the topological characteristics of the integrated circuit. ('432 patent at Col. 1, lines 38-44; Col. 2, lines 44-49; Abstract at lines 19-23). Thus, based on claims 13-20 and the '432 patent's description it is beyond dispute that claims 13-20's processes are not processes for "directly manufacturing" ASICs as required for an infringement claim pursuant to Section 271(g).

Despite this, Ricoh will likely argue that the '432 patent's processes for generating "netlists" and "mask data" are required and/or necessary for manufacturing ASICs. But that does not mean that these processes are "steps in the manufacture" of ASICs as required by Section 271(g). The Federal Circuit has recently held that "made by" in that section requires that "the process must be used directly in the manufacture of the product, and not merely as a predicate process to identify the product to be manufactured." *Housey*, 340 F. 3d at 1378. Here the processes of claims 13-20, like the process in *Housey*, "are not steps in the manufacture" of ASICs but instead are merely "predicate processes" that generate data identifying the product to be made. *See id*. at 1377.

D. Section 271(g) Is Limited To Processes Used *Directly* in the Manufacture of Physical Products, And Does Not Encompass "Predicate" Processes

In August of last year, the Federal Circuit construed the meaning of the phrase "a product which is made by a process" to require the manufacture of a physical product by an alleged infringer, and in doing so obviated Ricoh's only alleged avenue of relief against the ASIC Defendants. In *Housey*, the Federal Circuit reviewed the dismissal by the Delaware District Court of *Housey's*

counterclaim of patent infringement, for failure to state a claim against Bayer under Section 271(g). In response to Bayer's declaratory relief action, *Housey* alleged that Bayer infringed its patented methods of screening compounds for their ability to inhibit or activate proteins in a cell. The result of the patented screening process was information used to identify and describe new drugs that could then be manufactured using that information. *See Housey*, 340 F. 3d at 1369-70.

The parties agreed that the scope of the counterclaim for infringement extended to both the importation of a drug identified by the patented process as a protein inhibitor or activator, as well as to the importation into or use in the United States of information generated by the patented process, *i.e.*, "knowledge and information reflecting the identification or characterization of a drug acquired from using the patented methods." *Id.* at 1370.

In addressing both theories of infringement, the court analyzed both the language and legislative history of Section 271(g) in detail, including a discussion of the enactment of the Process Patents Amendments Act, drafted in response to concerns that competitors could avoid infringement of method patents by employing those methods abroad, and then importing the resulting products into the United States. *Id.* at 1373-74. The Act supplemented existing remedies available from the International Trade Commission ("ITC") under 19 U.S.C. § 1337 (including in its definition of "unfair methods of competition" the importation into or the sale within the United States of articles made by means of a process covered by a United States patent). *See id.* (citing 19 U.S.C. § 1337 (a)(1)(B)). The court's analysis of the phrase "product which is made by a process," along with its extensive review of the legislative history, led the Federal Circuit to conclude: "that Congress was concerned solely with physical goods that had undergone manufacture." *Id.* The court noted, "[e]ach and every reference to the provision that became Section 271(g) describes it as directed to manufacturing." *Id.* at 1374. Accordingly, the Federal Circuit held that Section 271(g) can apply only to the importation or sale of "physical objects" from manufacturing processes, and that "the production of information is not within the scope of processes of 'manufacture.'" *Id.* at 1376-77.

As indicated above, the Federal Circuit's analysis, in the *Housey* decision, did not end with the Court's holding that information is not a "product" under Section 271(g). Because *Housey's* claim of infringement against Bayer also included actual drug products that *Housey* alleged Bayer had

manufactured using the process described in the patent-in-suit, the Court went on to assess the merits of those assertions as well. *See Housey*, 340 F.3d at 1376-77. Neither the court nor the parties disputed the fact that such drugs were physical products that had been manufactured. *See id.* at 1377. Here Ricoh has similarly asserted that the ASIC Defendants are infringing its process claims via the importation, use, or sale of ASICs. (Complaint at ¶¶ 16, 22, 28, 34, 40, and 46). The ASIC Defendants do not dispute that an ASIC indeed is also a manufactured product. The issue here, however, as it was in *Housey*, is the "necessary relationship between the 'process patented in the United States' and the resulting product[.]" *Housey*, 340 F.3d at 1377.

In determining whether a drug "identified as useful through the use of a patented process" was a product made by that process under Section 271(g), the *Housey* court observed that it was charged with resolving the "critical question of proximity to the product of the patented process" on a case-by-case basis. *Id.* at 1377 (quoting *Bio-Technology General Corp. v. Genetech, Inc.*, 80 F.3d 1553, 1561 (Fed. Cir. 1996)). To assess this requisite "proximity," the Federal Circuit turned once again to the plain language of the statute, noting that it required the allegedly infringing product to have been "made *by* a process patented in the United States." *Id.* at 1377-78 (quoting 35 U.S.C. § 271(g)) (emphasis in original). The Court interpreted the word "by" to require that the process be used *directly* in the manufacture of the product, "and not merely as a predicate process to identify the product to be manufactured." *Id.* at 1378. Accordingly, because Bayer did not use the patented process in the actual manufacturing of the drug, the Court held that that drug was not a product "made by" those processes—regardless of the fact that Bayer had been able to study the compound and generate information regarding its characteristics using the claimed processes. *Id.*

Ricoh's Infringement Allegations	The *Housey* Decision
The "application specific integrated circuits" (ASICs) are physical products. (Complaint at ¶¶ 16, 22, 28, 34, 40, and 46).	A drug is "a physical product that has been manufactured." *Id*. at 1377.
Claims 13-20 are all directed to a "design process for designing an application specific integrated circuit" (ASIC). ('432 patent at Col. 16, lines 34-35, 66; Col. 17, lines 1, 4, 8, 11-12; Col. 18, lines 15, 22).	"The disclosed method provides a process for identifying the effect that different agents have on the activity of the suspect protein"—i.e., the protein linked to a disease. *Id*. at 1369.
The use of claims 13, 15-20's processes result in a netlist—i.e., a "list identifying the hardware cells and their interconnection requirements." ('432 patent at Abstract at lines 17-18; Col. 2, lines 43-45; Col. 1, lines 38-40). Claim 14's process uses the netlist generated by claim 13 to generate mask data—i.e., geometrical information providing the physical layout level description of the topological characteristics of the integrated circuit. ('432 patent at Col. 1, lines 38-44; Col. 2, lines 44-49; Abstract at lines 19-23).	The use of the patented process results in the identification and generation of data for identifying a useful drug. *Id*. at 1378.
The mask data is used <u>once</u> to make the masks. These masks are used <u>repeatedly</u> by and with other processes to manufacture the ASICs. Therefore, the processes for generating netlists and mask data are not steps in the manufacture of a final ASIC.	The data identifying the agent with a useful effect is generated once. Other processes to manufacture the drug then use the information. Therefore, the "processes of identification and generation of data are not steps in the manufacture of a final drug product." *Id*. at 1378.

Likewise, as illustrated in the above table, the processes Ricoh claims under the '432 patent are not, and cannot be, used *directly* in the manufacture of integrated circuits (ASICs). Instead, and as discussed in detail above, such processes result in the generation of design information only. Indeed, as in *Housey*, such processes are merely predicates for the identification of the product to be manufactured. *See id.* at 1378. More specifically, the processes of claims 13-20 generate "netlists" (claims 13, 15-20) and "mask data" (claim 14). This design information is used <u>once</u> to produce the masks. It is these masks that are used repeatedly by the processes used directly to manufacture the ASICs. This case fits squarely within the Federal Circuit's analysis. An ASIC is not a product "made by" any process described in the '432 patent, and the process that *is* contemplated by the '432 patent

merely generates "design information," which is not a manufactured product as required by Section 271(g). The table above illustrates that the Federal Circuit's *Housey* decision cannot be distinguished from Ricoh's infringement allegations here.

The computer-aided design methods claimed in Ricoh's '432 patent do not describe the steps of any process for the manufacture of a physical ASIC, but produce only information used in the process of *designing* a product. Tellingly, Ricoh's complaint does not even allege that the ASIC Defendants use Ricoh's patented method in the *manufacture* of their ASICs. Instead Ricoh alleges that they sell ASICs "designed by or using information generated by," the claimed process. (*See* Complaint ¶¶ 28, 34). These allegations are not sufficient to make out a cause of action under Section 271(g). Moreover, amending the complaint to correct this defect would be futile, since the process claimed in Ricoh's patent is, at best, a precursor or predicate process to the manufacture of physical goods, and cannot, as a matter of law, be enforced under Section 271(g). Therefore, judgment on the pleadings is proper, and the Court should grant this Motion on behalf of the ASIC Defendants.

V. CONCLUSION

As detailed above, none of the ASIC Defendants has been, or can be, accused of manufacturing a "physical product" that is "made by" the patented processes of the '432 patent. The process set forth in the '432 patent produces information identifying designs for integrated circuits—not the chips themselves or any other physical product. As such, and in accordance with the Federal Circuit's analysis and construction of Section 271(g), relief is simply not available to a patentee—such as Ricoh—where there are no "physical goods that are manufactured *directly* by the patented process." Accordingly, as a matter of law, none of the ASIC Defendants or anyone else can infringe the '432's patented processes under Section 271(g), and the Court should enter judgment on the pleadings dismissing Ricoh's infringement complaint.

Dated: February 10, 2003

Respectfully submitted,

HOWREY SIMON ARNOLD & WHITE, LLP

By: /s/ Thomas Mavrakakis
Teresa M. Corbin
Attorneys for Defendants
AEROFLEX INCORPORATED, AMI SEMICONDUCTOR, INC., MATROX ELECTRONIC SYSTEMS, LTD., MATROX GRAPHICS INC., MATROX INTERNATIONAL CORP., and MATROX TECH, INC.

COPY

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

RICOH COMPANY, LTD. Plaintiff, v. AEROFLEX INCORPORATED, AMI SEMICONDUCTOR, INC., MATROX ELECTRONIC SYSTEMS LTD., MATROX GRAPHICS INC., MATROX INTERNATIONAL CORP. and MATROX TECH, INC., Defendants.	)	C.A. No. 03-103

COMPLAINT

Plaintiff Ricoh Company, Ltd. ("Ricoh") for its Complaint against Defendants Aeroflex Incorporated ("Aeroflex"), AMI Semiconductor, Inc. ("AMI"), Matrox Electronic Systems Ltd. ("Matrox"), Matrox Graphics Inc. ("Matrox Graphics"), Matrox International Corp. ("Matrox Int'l"), and Matrox Tech, Inc. ("Matrox Tech"), alleges as follows:

PARTIES

1. Plaintiff Ricoh is a corporation organized under the laws of Japan and maintains its principal place of business at 3-6 1-chome, Nakamagome, Tokyo, Japan.

2. Upon information and belief, Defendant Aeroflex is a corporation organized under the laws of the State of Delaware, maintains its principal place of business at 35 S. Service Road, Plainview, NY, 11803, and has appointed The Corporation Trust Company, Corporation Trust Center, 1209 Orange Street, Wilmington, DE 19801 as its registered agent in Delaware.

3. Upon information and belief, Defendant AMI is a corporation organized under the laws of the State of Delaware, maintains its principal place of business at 2300 Buckskin Road,

Pocatello, ID 83201, and has appointed The Corporation Trust Company, Corporation Trust Center, 1209 Orange Street, Wilmington, DE 19801 as its registered agent in Delaware.

4. Upon information and belief, Defendant Matrox is a corporation organized under the laws of Quebec, Canada, maintains its principal place of business at 1055 Boul St-Regis, Dorval, Quebec H9P 2T4 Canada and is doing business in Delaware and/or has committed the acts complained of in Delaware.

5. Upon information and belief, Defendant Matrox Graphics is a corporation organized under the laws of Quebec, Canada, maintains its principal place of business at 1055 Boul St-Regis, Dorval, Quebec H9P 2T4 Canada and is doing business in Delaware and/or has committed the acts complained of in Delaware.

6. Upon information and belief, Defendant Matrox Int'l is a corporation organized under the laws of New York, maintains its principal place of business at 625 State Rt 3, Unit B, Plattsburgh, NY 12901, and is doing business in Delaware and/or has committed the acts complained of in Delaware.

7. Upon information and belief, Defendant Matrox Tech is a corporation organized under the laws of the State of Delaware, maintains its principal place of business at 1075 Broken Sound Parkway, NW, Boca Raton, FL 33487-3524 and has appointed The Corporation Trust Company, Corporation Trust Center, 1209 Orange Street, Wilmington, DE 19801 as its registered agent in Delaware.

JURISDICTION

8. This action arises under the patent laws of the United States, Title 35, United States Code, and more particularly under 35 U.S.C. §§ 271 et. seq.

9. This Court has subject matter jurisdiction over this patent infringement action under the Judicial Code of the United States, 28 U.S.C. §§ 1338(a) and 1331.

10. This Court has personal jurisdiction over the Defendants because Defendants are present and/or doing business in Delaware either directly or through their agents, or alternatively, are incorporated in Delaware.

VENUE

11. Venue is proper in this district pursuant to 28 U.S.C. § 1391 in that Defendants reside in this judicial district and/or a substantial part of the events or omissions giving rise to the claim occurred in this judicial district and/or are found in this judicial district and/or are aliens.

FACTUAL BACKGROUND

12. On May 1, 1990, the U.S. Patent and Trademark Office ("USPTO") duly and legally issued United States Letters Patent No. 4,922,432 (the " '432 Patent") in the names of Hideaki Kobayashi and Masahiro Shindo for their invention titled "Knowledge Based Method and Apparatus for Designing Integrated Circuits using Functional Specifications." A copy of the '432 Patent is attached hereto as Exhibit 1.

13. By assignment, Ricoh is the owner of the entire right, title, and interest in the '432 Patent and has the sole right to sue and recover for infringement thereof.

14. The '432 Patent describes, inter alia, a method for designing an application specific integrated circuit. By using the invention of the '432 Patent, one can define functional architecture independent specifications for an integrated circuit and translate functional architecture independent specifications into the detailed information needed for directly producing the integrated circuit.

PATENT INFRINGEMENT

COUNT 1

15. Ricoh repeats and realleges the allegations set forth in paragraphs 1 through 14 hereof.

16. Upon information and belief, Aeroflex has been and is now infringing the '432 Patent by using, offering to sell, and/or by selling and/or importing into the United States application specific integrated circuits designed by or using information generated by, the process of one or more of claims 13-20 of the '432 Patent, either literally or under the doctrine of equivalents.

17. Upon information and belief, Aeroflex will continue to infringe the '432 Patent unless enjoined by this Court.

18. As a consequence of Aeroflex's infringement, Ricoh has been irreparably damaged to an extent not yet determined, and Ricoh will continue to be irreparably damaged by such acts in the future unless Aeroflex is enjoined by this Court from committing further acts of infringement.

19. Upon information and belief, Aeroflex's infringement of the '432 Patent is willful.

20. Ricoh is entitled to recover damages adequate to compensate for Aeroflex's infringement.

COUNT 2

21. Ricoh repeats and realleges the allegations set forth in paragraphs 1 through 14 hereof.

22. Upon information and belief, AMI has been and is now infringing the '432 Patent by using, offering to sell, and/or by selling and/or importing into the United States application

specific integrated circuits designed by or using information generated by, the process of one or more of claims 13-20 of the '432 Patent, either literally or under the doctrine of equivalents.

23. Upon information and belief, AMI will continue to infringe the '432 Patent unless enjoined by this Court.

24. As a consequence of AMI's infringement, Ricoh has been irreparably damaged to an extent not yet determined, and Ricoh will continue to be irreparably damaged by such acts in the future unless AMI is enjoined by this Court from committing further acts of infringement.

25. Upon information and belief, AMI's infringement of the '432 Patent is willful.

26. Ricoh is entitled to recover damages adequate to compensate for AMI's infringement.

COUNT 3

27. Ricoh repeats and realleges the allegations set forth in paragraphs 1 through 14 hereof.

28. Upon information and belief, Matrox has been and is now infringing the '432 Patent by using, offering to sell, and/or by selling and/or importing into the United States application specific integrated circuits designed by or using information generated by, the process of one or more of claims 13-20 of the '432 Patent, either literally or under the doctrine of equivalents.

29. Upon information and belief, Matrox will continue to infringe the '432 Patent unless enjoined by this Court.

30. As a consequence of Matrox's infringement, Ricoh has been irreparably damaged to an extent not yet determined, and Ricoh will continue to be irreparably damaged by such acts

in the future unless Matrox is enjoined by this Court from committing further acts of infringement.

31. Upon information and belief, Matrox's infringement of the '432 Patent is willful.

32. Ricoh is entitled to recover damages adequate to compensate for Matrox's infringement.

COUNT 4

33. Ricoh repeats and realleges the allegations set forth in paragraphs 1 through 14 hereof.

34. Upon information and belief, Matrox Graphics has been and is now infringing the '432 Patent by using, offering to sell, and/or by selling and/or importing into the United States application specific integrated circuits designed by or using information generated by, the process of one or more of claims 13-20 of the '432 Patent, either literally or under the doctrine of equivalents.

35. Upon information and belief, Matrox Graphics will continue to infringe the '432 Patent unless enjoined by this Court.

36. As a consequence of Matrox Graphics' infringement, Ricoh has been irreparably damaged to an extent not yet determined, and Ricoh will continue to be irreparably damaged by such acts in the future unless Matrox Graphics is enjoined by this Court from committing further acts of infringement.

37. Upon information and belief, Matrox Graphics' infringement of the '432 Patent is willful.

38. Ricoh is entitled to recover damages adequate to compensate for Matrox Graphics' infringement.

COUNT 5

39. Ricoh repeats and realleges the allegations set forth in paragraphs 1 through 14 hereof.

40. Upon information and belief, Matrox Int'l has been and is now infringing the '432 Patent by using, offering to sell, and/or by selling and/or importing into the United States application specific integrated circuits designed by or using information generated by, the process of one or more of claims 13-20 of the '432 Patent, either literally or under the doctrine of equivalents.

41. Upon information and belief, Matrox Int'l will continue to infringe the '432 Patent unless enjoined by this Court.

42. As a consequence of Matrox Int'l's infringement, Ricoh has been irreparably damaged to an extent not yet determined, and Ricoh will continue to be irreparably damaged by such acts in the future unless Matrox Int'l is enjoined by this Court from committing further acts of infringement.

43. Upon information and belief, Matrox Int'l's infringement of the '432 Patent is willful.

44. Ricoh is entitled to recover damages adequate to compensate for Matrox Int'l's infringement.

COUNT 6

45. Ricoh repeats and realleges the allegations set forth in paragraphs 1 through 14 hereof.

46. Upon information and belief, Matrox Tech has been and is now infringing the '432 Patent by using, offering to sell, and/or by selling and/or importing into the United States

application specific integrated circuits designed by or using information generated by, the process of one or more of claims 13-20 of the '432 Patent, either literally or under the doctrine of equivalents.

47. Upon information and belief, Matrox Tech will continue to infringe the '432 Patent unless enjoined by this Court.

48. As a consequence of Matrox Tech's infringement, Ricoh has been irreparably damaged to an extent not yet determined, and Ricoh will continue to be irreparably damaged by such acts in the future unless Matrox Tech is enjoined by this Court from committing further acts of infringement.

49. Upon information and belief, Matrox Tech's infringement of the '432 Patent is willful.

50. Ricoh is entitled to recover damages adequate to compensate for Matrox Tech's infringement.

PRAYER FOR RELIEF

WHEREFORE, Ricoh prays for entry of judgment:

A. that Aeroflex has infringed the '432 Patent;

B. that Aeroflex, its agents, employees, representatives, successors, and assigns and those acting, or purporting to act, in privity or in concert with Aeroflex, be preliminarily and permanently enjoined from further infringement of the '432 Patent;

C. that Aeroflex account for and pay to Ricoh all damages under 35 U.S.C. § 284, including enhanced damages, caused by the infringement of the '432 Patent, and attorneys' fees pursuant to 35 U.S.C. § 285;

D. that Ricoh be granted pre-judgment and post-judgment interest on the damages

caused to it by reason of Aeroflex's infringement of the '432 Patent;

E. that AMI has infringed the '432 Patent;

F. that AMI, its agents, employees, representatives, successors, and assigns and those acting, or purporting to act, in privity or in concert with AMI, be preliminarily and permanently enjoined from further infringement of the '432 Patent;

G. that AMI account for and pay to Ricoh all damages under 35 U.S.C. § 284, including enhanced damages, caused by the infringement of the '432 Patent, and attorneys' fees pursuant to 35 U.S.C. § 285;

H. that Ricoh be granted pre-judgment and post-judgment interest on the damages caused to it by reason of AMI's infringement of the '432 Patent;

I. that Matrox has infringed the '432 Patent;

J. that Matrox, its agents, employees, representatives, successors, and assigns and those acting, or purporting to act, in privity or in concert with Matrox, be preliminarily and permanently enjoined from further infringement of the '432 Patent;

K. that Matrox account for and pay to Ricoh all damages under 35 U.S.C. § 284, including enhanced damages, caused by the infringement of the '432 Patent, and attorneys' fees pursuant to 35 U.S.C. § 285;

L. that Ricoh be granted pre-judgment and post-judgment interest on the damages caused to it by reason of Matrox's infringement of the '432 Patent;

M. that Matrox Graphics has infringed the '432 Patent;

N. that Matrox Graphics, its agents, employees, representatives, successors, and assigns and those acting, or purporting to act, in privity or in concert with Matrox Graphics, be preliminarily and permanently enjoined from further infringement of the '432 Patent;

O. that Matrox Graphics account for and pay to Ricoh all damages under 35 U.S.C. § 284, including enhanced damages, caused by the infringement of the '432 Patent, and attorneys' fees pursuant to 35 U.S.C. § 285;

P. that Ricoh be granted pre-judgment and post-judgment interest on the damages caused to it by reason of Matrox Graphics' infringement of the '432 Patent;

Q. that Matrox Int'l has infringed the '432 Patent;

R. that Matrox Int'l, its agents, employees, representatives, successors, and assigns and those acting, or purporting to act, in privity or in concert with Matrox Int'l, be preliminarily and permanently enjoined from further infringement of the '432 Patent;

S. that Matrox Int'l account for and pay to Ricoh all damages under 35 U.S.C. § 284, including enhanced damages, caused by the infringement of the '432 Patent, and attorneys' fees pursuant to 35 U.S.C. § 285;

T. that Ricoh be granted pre-judgment and post-judgment interest on the damages caused to it by reason of Matrox Int'l's infringement of the '432 Patent;

U. that Matrox Tech has infringed the '432 Patent;

V. that Matrox Tech, its agents, employees, representatives, successors, and assigns and those acting, or purporting to act, in privity or in concert with Matrox Tech, be preliminarily and permanently enjoined from further infringement of the '432 Patent;

W. that Matrox Tech account for and pay to Ricoh all damages under 35 U.S.C. § 284, including enhanced damages, caused by the infringement of the '432 Patent, and attorneys' fees pursuant to 35 U.S.C. § 285;

X. that Ricoh be granted pre-judgment and post-judgment interest on the damages caused to it by reason of Matrox Tech's infringement of the '432 Patent;

Y. that costs be awarded to Ricoh; and

Z. that Ricoh be granted such other and further relief as the Court may deem just and proper under the current circumstances.

Robert W. Whetzel by SJF/dmw (4025)

Robert W. Whetzel (#2288)
Steven J. Fineman (#4025)
Richards, Layton & Finger, P.A.
One Rodney Square
Post Office Box 551
Wilmington, Delaware 19899
(302) 651-7700
Attorneys for Plaintiff

OF COUNSEL:
Gary M. Hoffman
Edward A. Meilman
Eric Oliver
Dickstein Shapiro Morin & Oshinsky LLP
2101 L Street, N.W.
Washington, D.C. 20037-1526
(202) 785-9700

Dated: January 21, 2003

United States Patent [19]

Kobayashi et al.

[11] **Patent Number: 4,922,432**

[45] **Date of Patent: May 1, 1990**

[54] **KNOWLEDGE BASED METHOD AND APPARATUS FOR DESIGNING INTEGRATED CIRCUITS USING FUNCTIONAL SPECIFICATIONS**

[75] Inventors: **Hideaki Kobayashi**, Columbia, S.C.; **Masahiro Shindo**, Osaka, Japan

[73] Assignees: **International Chip Corporation**, Columbia, S.C.; **Ricoh Company, Ltd.**, Tokyo, Japan

[21] Appl. No.: 143,821

[22] Filed: Jan. 13, 1988

[51] Int. Cl.[5] G06F 15/60

[52] U.S. Cl. 364/490; 364/489; 364/488; 364/521

[58] Field of Search 364/488–491, 364/521, 300, 513

[56] References Cited

U.S. PATENT DOCUMENTS

4,635,208	1/1987	Coleby et al.	364/491
4,638,442	1/1987	Bryant et al.	364/489
4,648,044	3/1987	Hardy et al.	364/513
4,651,284	3/1987	Watanabe et al.	364/491
4,656,603	4/1987	Dunn	364/488
4,658,370	4/1987	Erman et al.	364/513
4,675,829	6/1987	Clemenson	364/513
4,700,317	10/1987	Watanabe et al.	364/521
4,703,435	10/1987	Darringer et al.	364/488
4,803,636	2/1989	Nishiyama et al.	364/491

FOREIGN PATENT DOCUMENTS

1445914	8/1976	United Kingdom	364/490

OTHER PUBLICATIONS

"Verifying Compiled Silicon", by E. K. cheng, VLSI Design, Oct. 1984, pp. 1–4.

"CAD System for IC Design", by M. E. Daniel et al., IEEE Trans. on Computer-Aided Design of Integrated Circuits & Systems, vol. CAD–1, No. 1, Jan. 1982, pp. 2–12.

"An Overview of Logic Synthesis System", by L. Trevillyan, 24th ACM/IEEE Design Automation Conference, 1978, pp. 166–172.

"Methods Used in an Automatic Logic Design Generator", by T. D. Friedman et al., IEEE Trans. on Computers, vol. C–18, No. 7, Jul. 1969, pp. 593–613.

"Experiments in Logic Synthesis", by J. A. Darringer, IEEE ICCC, 1980.

"A Front End Graphic Interface to First Silicon Compiler", by J. H. Nash, EDA 84, Mar. 1984.

"quality of Designs from An Automatic Logic Generator", by T. D. Friedman et al., IEEE 7th DA Conference, 1970, pp. 71–89.

"A New Look at Logic Synthesis", by J. A. Darringer et al., IEEE 17th D. A. Conference 1980, pp. 543–548.

Trevillyan—Trickey, H., Flamel: *A High Level Hardward Compiler*, IEEE Transactions On Computer Aided Design, Mar. 1987, pp. 259–269.

Parker et al., The *CMU Design Automation System—An Example of Automated Data Path Design*, Proceedings Of The 16th Design Automation Conference, Las Vegas, Nev., 1979, pp. 73–80.

An Engineering Approach to Digital Design, William I. Fletcher, Prentice-Hall, Inc., pp. 491–505.

Primary Examiner—Felix D. Gruber

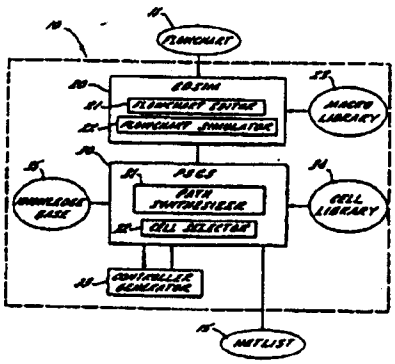
Assistant Examiner—V. N. Trans

Attorney, Agent, or Firm—Bell, Seltzer, Park & Gibson

[57] ABSTRACT

The present invention provides a computer-aided design system and method for designing an application specific integrated circuit which enables a user to define functional architecture independent specifications for the integrated circuit and which translates the functional architecture independent specifications into the detailed information needed for directly producing the integrated circuit. The functional architecture independent specifications of the desired integrated circuit can be defined at the functional architecture independent level in a flowchart format. From the flowchart, the system and method uses artificial intelligence and expert systems technology to generate a system controller, to select the necessary integrated circuit hardware cells needed to achieve the functional specifications, and to generate data and control paths for operation of the integrated circuit. This list of hardware cells and their interconnection requirements is set forth in a netlist. From the netlist it is possible using known manual techniques or existing VLSI CAD layout systems to generate the detailed chip level topological information (mask data) required to produce the particular application specific integrated circuit.

20 Claims, 12 Drawing Sheets



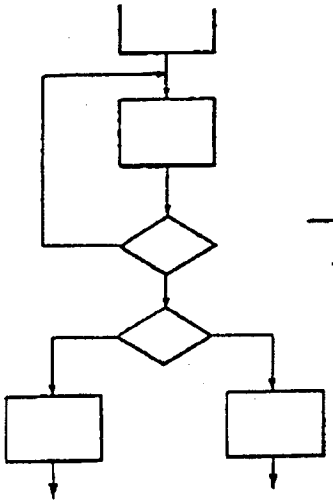


Fig. 1a.
FUNCTIONAL LEVEL

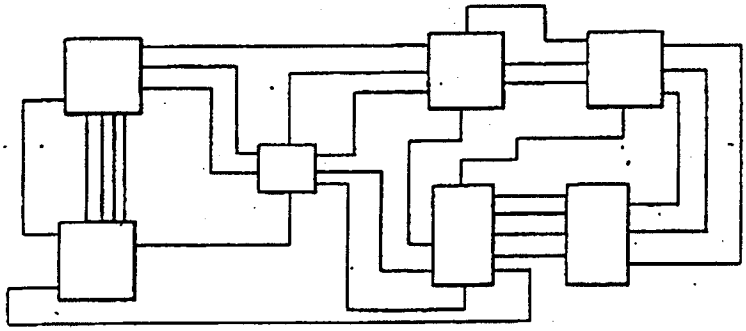


Fig. 1b.
STRUCTURAL LEVEL

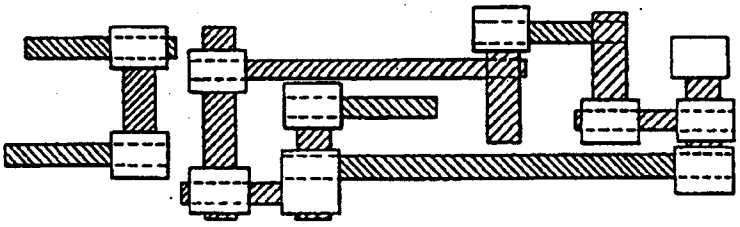


Fig. 1c.
PHYSICAL LAYOUT LEVEL

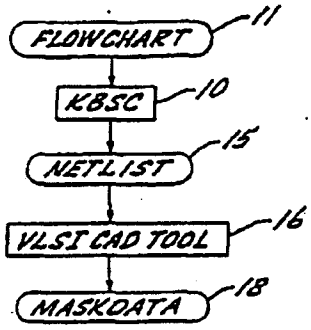

FLOWCHART
11
KBSC
10
NETLIST
15
VLSI CAD TOOL
16
MASKDATA
18

Fig. 2.

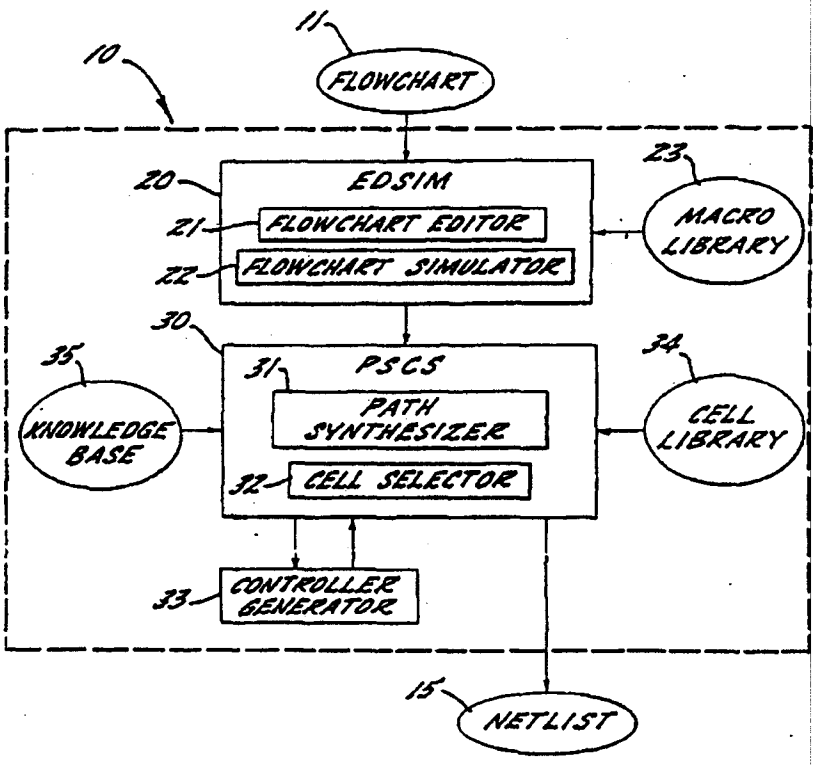

11
FLOWCHART
10
EDSIM
20
FLOWCHART EDITOR
21
FLOWCHART SIMULATOR
22
23
MACRO LIBRARY
30
PSCS
31
PATH SYNTHESIZER
CELL SELECTOR
32
35
KNOWLEDGE BASE
34
CELL LIBRARY
CONTROLLER GENERATOR
33
15
NETLIST

Fig. 3.

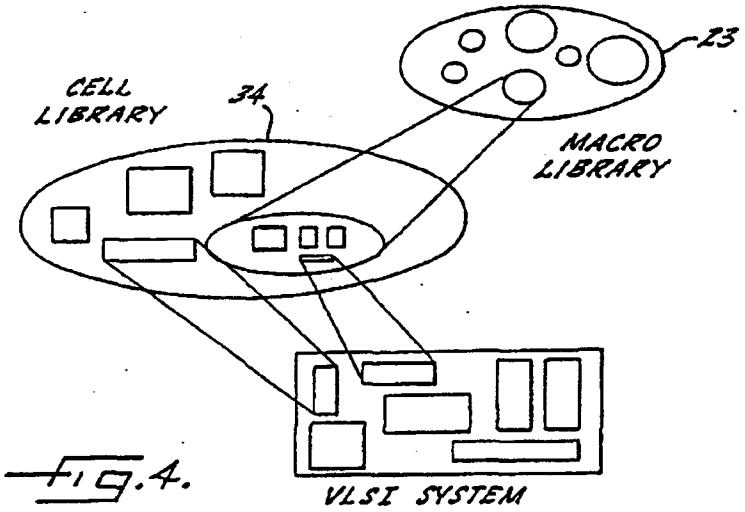

23
CELL LIBRARY
34
MACRO LIBRARY
VLSI SYSTEM

Fig. 4.

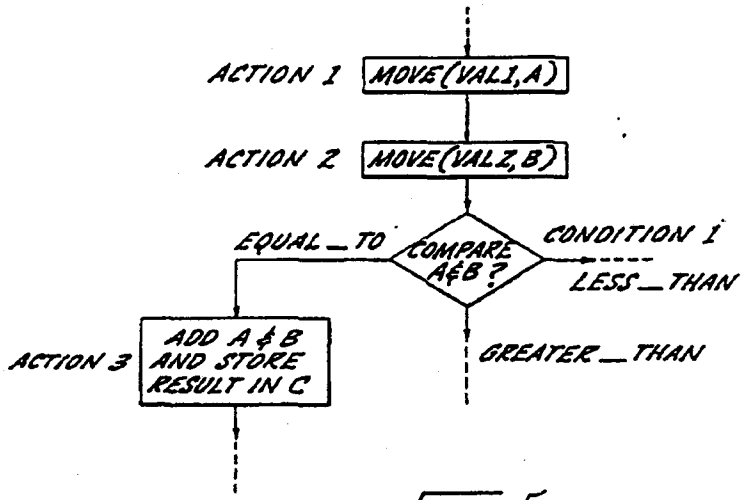

ACTION 1
MOVE(VAL1,A)
ACTION 2
MOVE(VAL2,B)
EQUAL_TO
COMPARE A&B?
CONDITION 1
LESS_THAN
GREATER_THAN
ACTION 3
ADD A & B AND STORE RESULT IN C

Fig. 5.

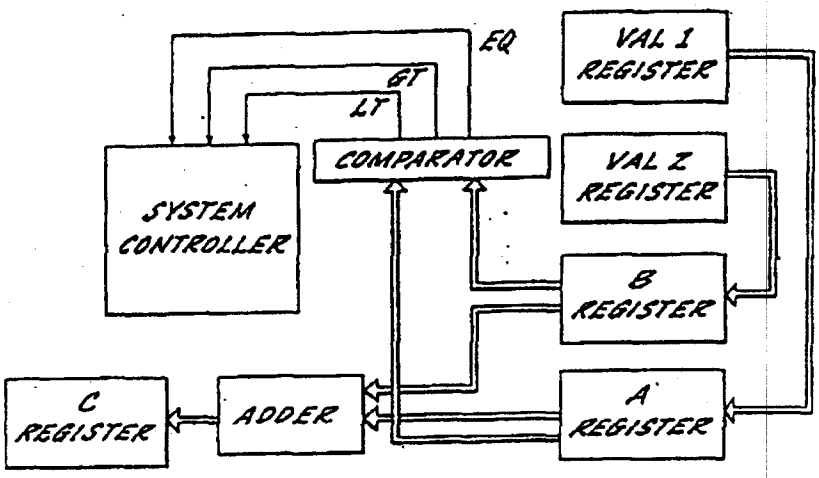

EQ
GT
LT
VAL 1 REGISTER
COMPARATOR
VAL 2 REGISTER
SYSTEM CONTROLLER
B REGISTER
C REGISTER
ADDER
A REGISTER

Fig. 6.

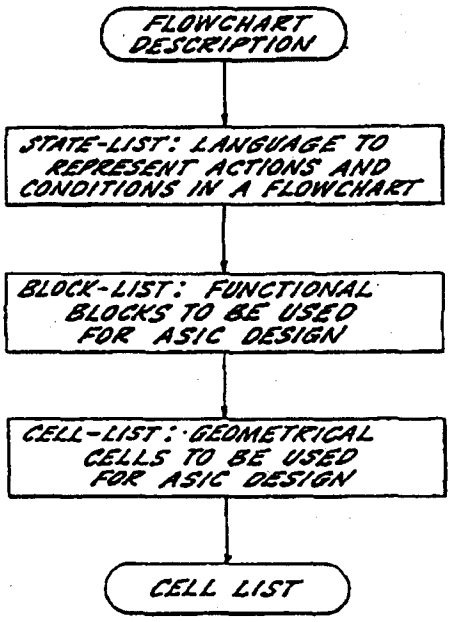

FLOWCHART DESCRIPTION
STATE-LIST: LANGUAGE TO REPRESENT ACTIONS AND CONDITIONS IN A FLOWCHART
BLOCK-LIST: FUNCTIONAL BLOCKS TO BE USED FOR ASIC DESIGN
CELL-LIST: GEOMETRICAL CELLS TO BE USED FOR ASIC DESIGN
CELL LIST

Fig. 9.

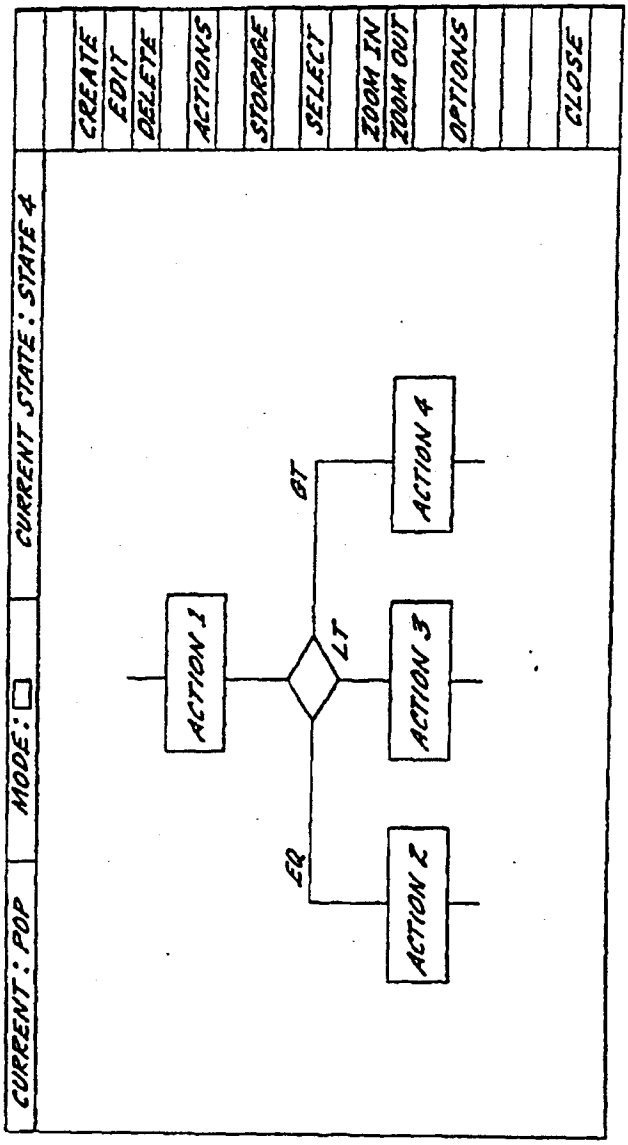

CREATE
EDIT
DELETE
ACTIONS
STORAGE
SELECT
ZOOM IN
ZOOM OUT
OPTIONS
CLOSE
CURRENT STATE: STATE 4
MODE:
CURRENT: POP
ACTION 1
GT
LT
EQ
ACTION 4
ACTION 3
ACTION 2

FIG. 7

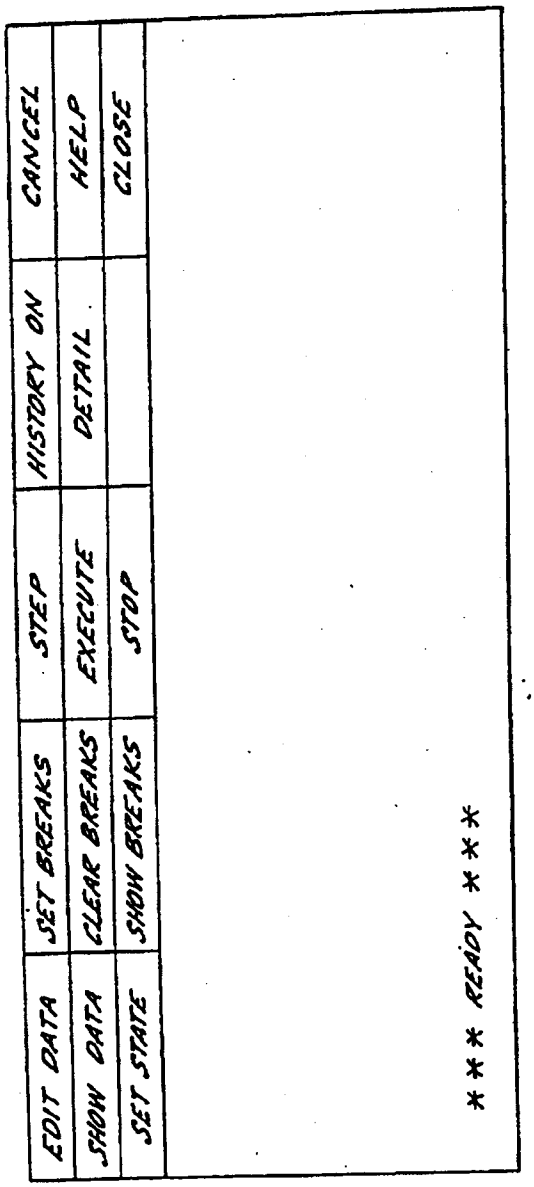

EDIT DATA
SHOW DATA
SET STATE
SET BREAKS
CLEAR BREAKS
SHOW BREAKS
STEP
EXECUTE
STOP
HISTORY ON
DETAIL
CANCEL
HELP
CLOSE
*** READY ***

FIG. 8.

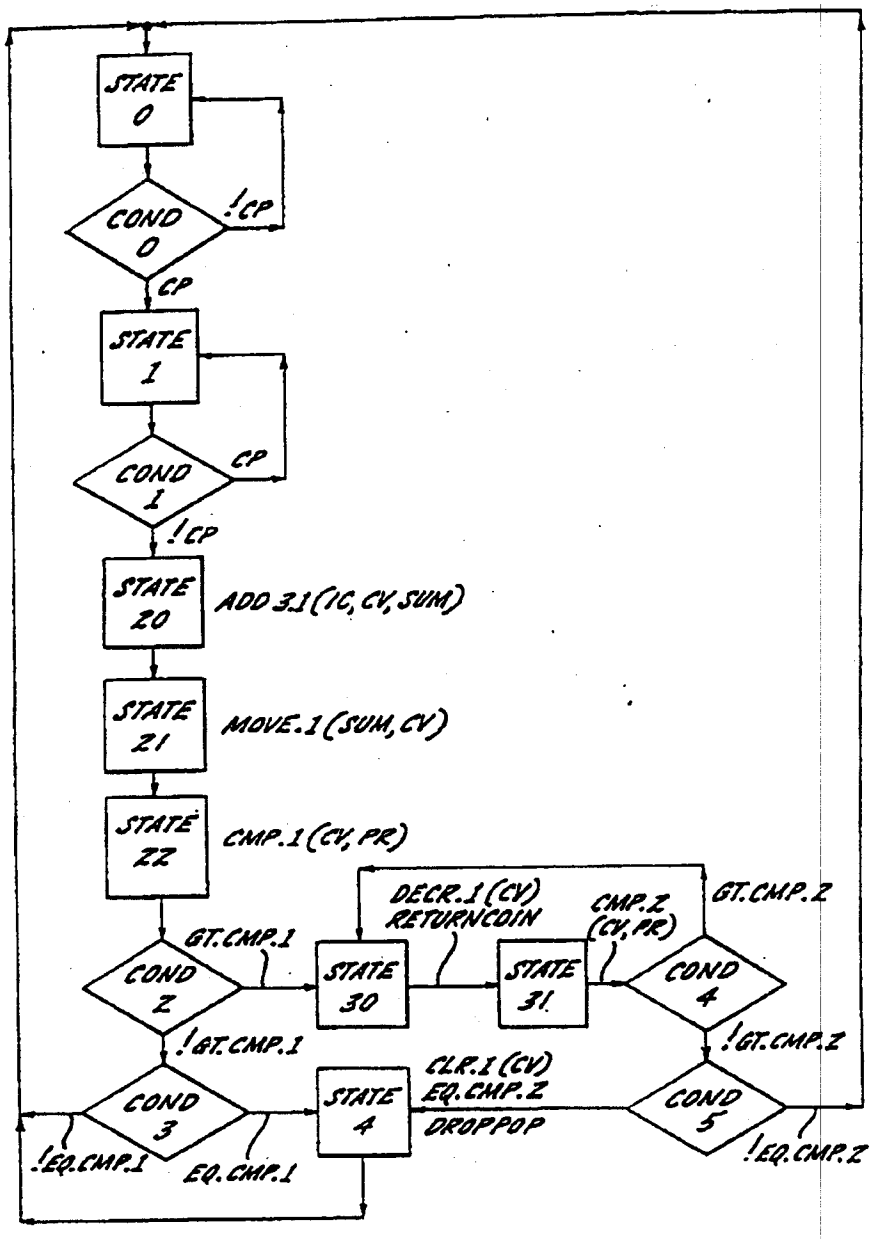

STATE 0
COND 0
!CP
CP
STATE 1
COND 1
CP
!CP
STATE 20
ADD.1(IC, CV, SUM)
STATE 21
MOVE.1(SUM, CV)
STATE 22
CMP.1(CV, PR)
GT.CMP.1
COND 2
!GT.CMP.1
STATE 30
DECR.1(CV)
RETURNCOIN
STATE 31
CMP.2 (CV, PR)
GT.CMP.2
COND 4
!GT.CMP.2
COND 3
!EQ.CMP.1
EQ.CMP.1
STATE 4
CLR.1(CV)
EQ.CMP.2
DROPPOP
COND 5
!EQ.CMP.2

Fig. 10.

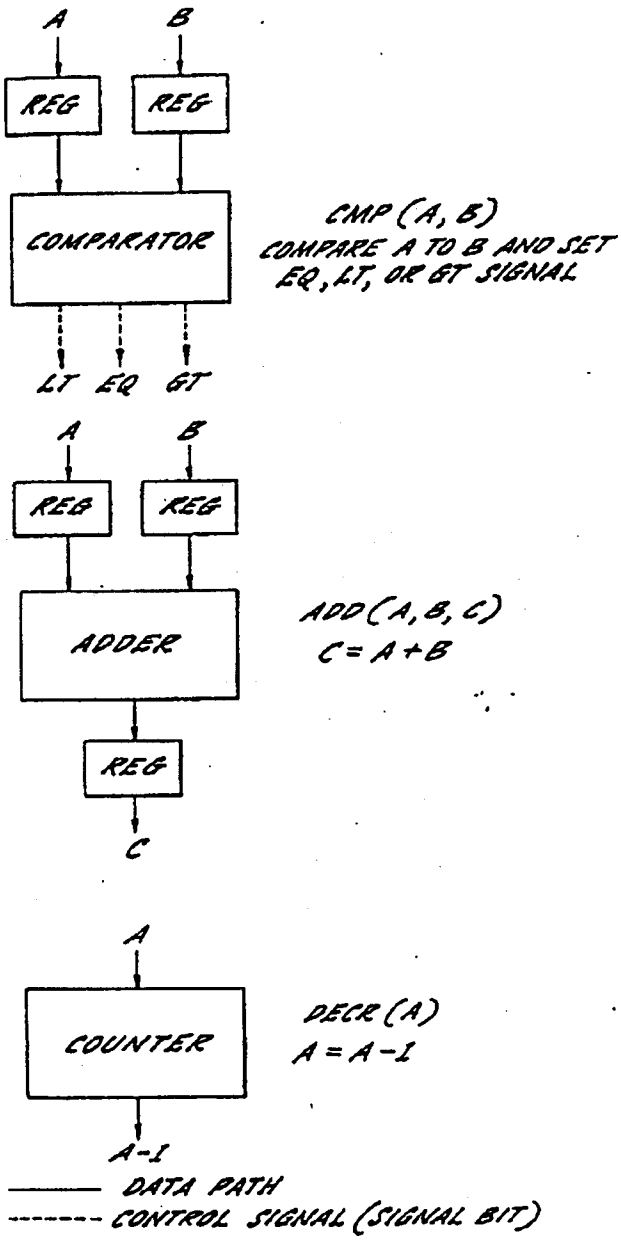

A
B
REG
REG
COMPARATOR
CMP (A, B)
COMPARE A TO B AND SET
EQ, LT, OR GT SIGNAL
LT EQ GT
A
B
REG
REG
ADDER
ADD (A, B, C)
C = A + B
REG
C
A
COUNTER
DECR (A)
A = A - 1
A-1
DATA PATH
CONTROL SIGNAL (SIGNAL BIT)

Fig. 11.

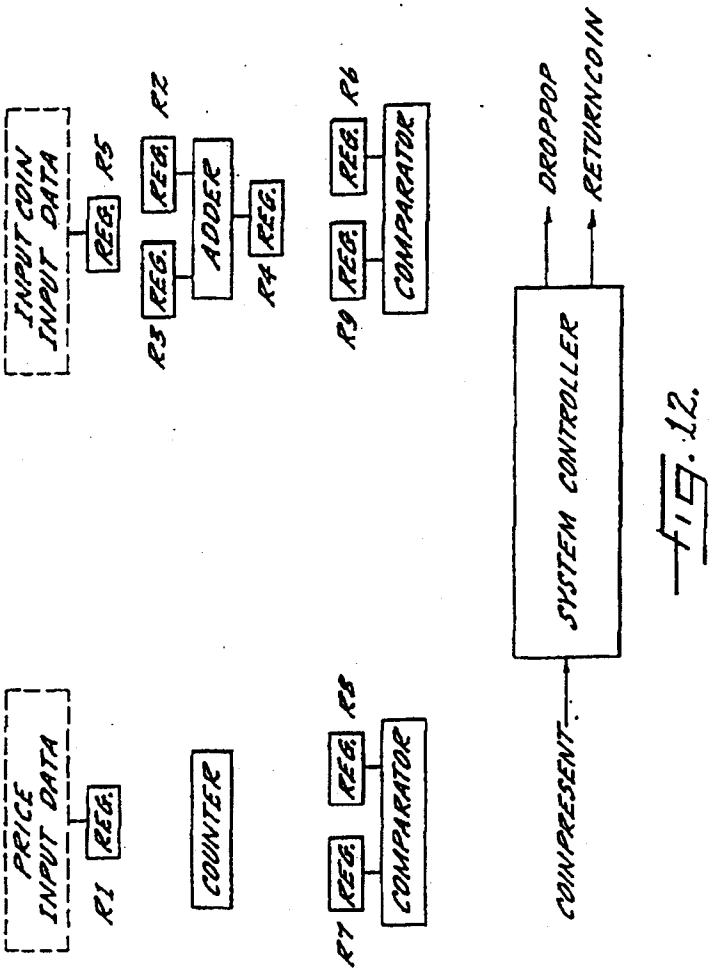

INPUT COIN
INPUT DATA
REG.
R5
REG.
R2
REG.
R3
ADDER
R4
REG.
REG.
R6
REG.
R9
COMPARATOR
DROPPOP
RETURNCOIN
SYSTEM CONTROLLER
COINPRESENT
PRICE
INPUT DATA
REG.
R1
COUNTER
REG.
R8
REG.
R7
COMPARATOR

FIG. 12.

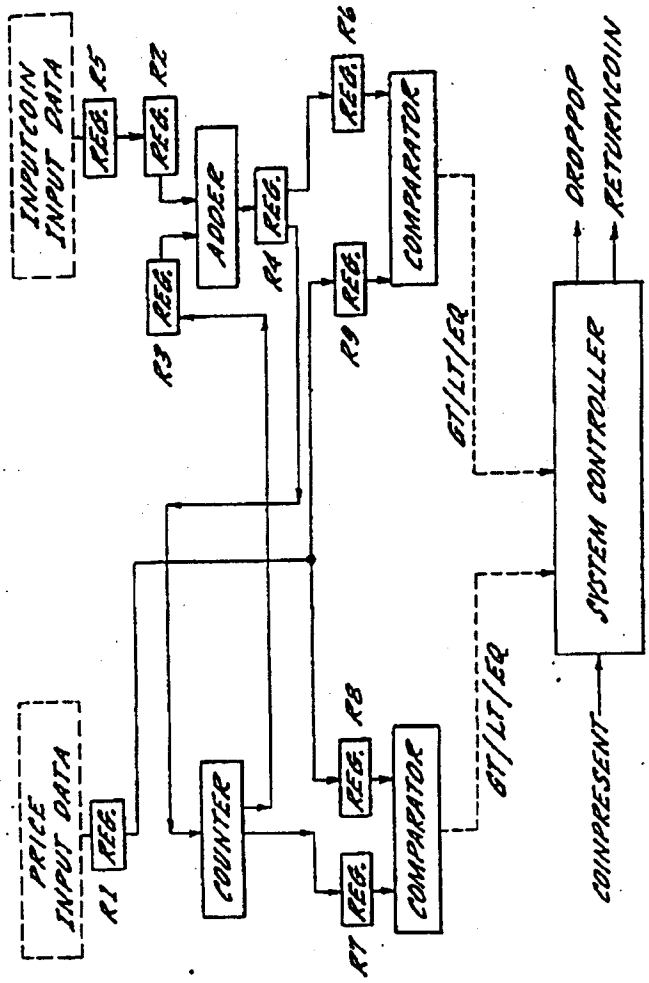

INPUT COIN
INPUT DATA
REG.
R5
R2
ADDER
R4
R6
COMPARATOR
R3
R9
GT/LT/EQ
DROP POP
RETURN COIN
SYSTEM CONTROLLER
PRICE
INPUT DATA
R1
COUNTER
R8
R7
COMPARATOR
GT/LT/EQ
COIN PRESENT

FIG. 13.

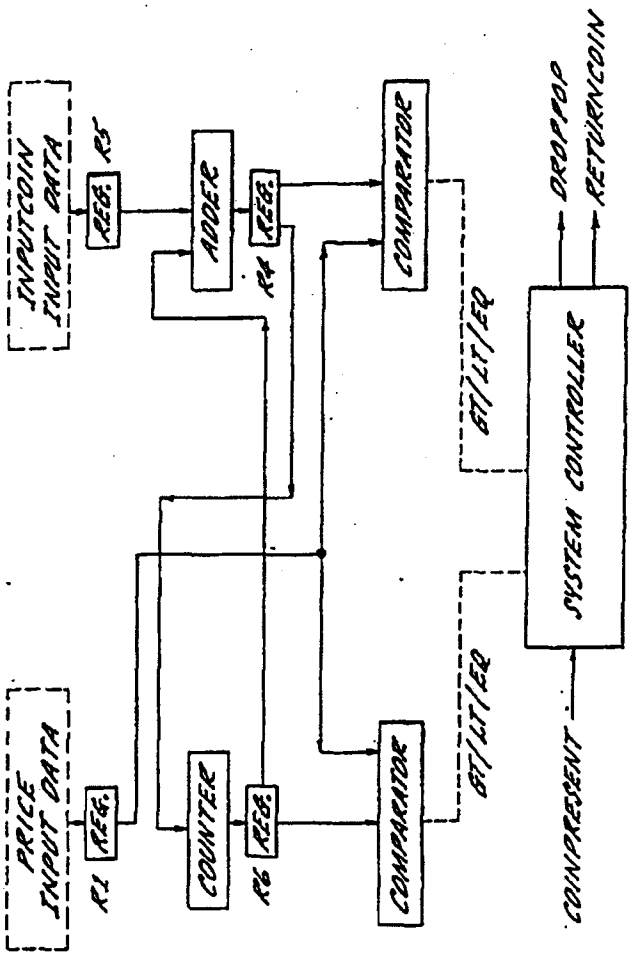

INVALID COIN INPUT DATA
REG.
R5
ADDER
REG.
R4
COMPARATOR
EQUAL
DROP COIN
RETURN COIN
SYSTEM CONTROLLER
PRICE INPUT DATA
REG.
R1
COUNTER
REG.
R6
COMPARATOR
EQUAL
COIN PRESENT

FIG. 14.

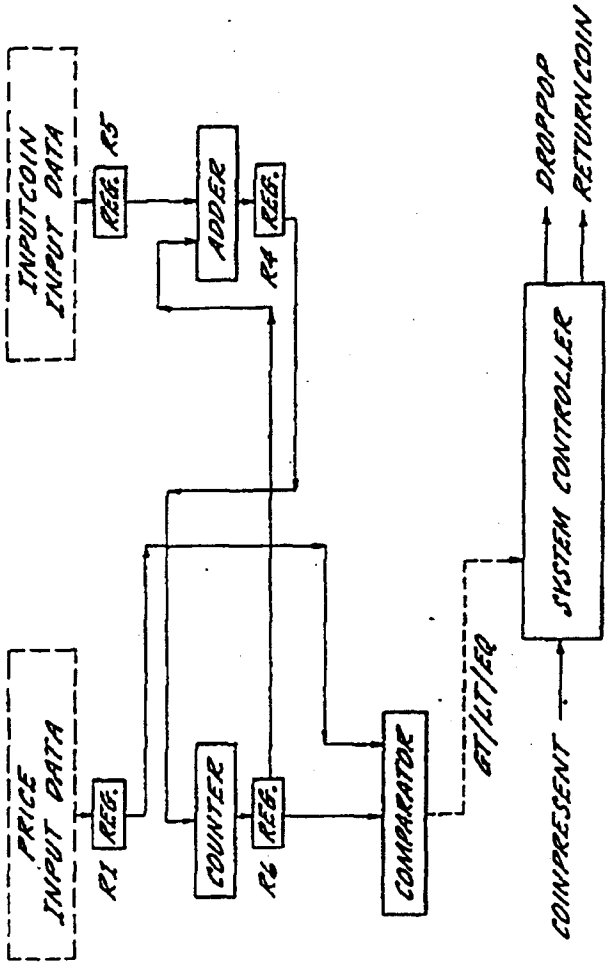

INPUT COIN
INPUT DATA
REG.
R5
ADDER
REG.
R4
PRICE
INPUT DATA
REG.
R1
COUNTER
REG.
R2
COMPARATOR
GT/LT/EQ
SYSTEM CONTROLLER
DROP POP
RETURN COIN
COIN PRESENT

FIG. 15.

KNOWLEDGE BASED METHOD AND APPARATUS FOR DESIGNING INTEGRATED CIRCUITS USING FUNCTIONAL SPECIFICATIONS

FIELD AND BACKGROUND OF THE INVENTION

This invention relates to the design of integrated circuits, and more particularly relates to a computer-aided method and apparatus for designing integrated circuits.

An application specific integrated circuit (ASIC) is an integrated circuit chip designed to perform a specific function, as distinguished from standard, general purpose integrated circuit chips, such as microprocessors, memory chips, etc. A highly skilled design engineer having specialized knowledge in VLSI circuit design is ordinarily required to design a ASIC. In the design process, the VLSI design engineer will consider the particular objectives to be accomplished and tasks to be performed by the integrated circuit and will create structural level design specifications which define the various hardware components required to perform the desired function, as well as the interconnection requirements between these components. A system controller must also be designed for synchronizing the operations of these components. This requires an extensive and all encompassing knowledge of the various hardware components required to achieve the desired objectives, as well as their interconnection requirements, signal level compatibility, timing compatibility, physical layout, etc. At each design step, the designer must do tedious analysis. The design specifications created by the VLSI design engineer may, for example, be in the form of circuit schematics, parameters or specialized hardware description languages (HDLs).

From the structural level design specifications, the description of the hardware components and interconnections is converted to a physical chip layout level description which describes the actual topological characteristics of the integrated circuit chip. This physical chip layout level description provides the mask data needed for fabricating the chip.

Due to the tremendous advances in very large scale integration (VLSI) technology, highly complex circuit systems are being built on a single chip. With their complexity and the demand to design custom chips at a faster rate, in large quantities, and for an ever increasing number of specific applications, computer-aided design (CAD) techniques need to be used. CAD techniques have been used with success in design and verification of integrated circuits, at both the structural level and at the physical layout level. For example, CAD systems have been developed for assisting in converting VLSI structural level descriptions of integrated circuits into the physical layout level topological mask data required for actually producing the chip. Although the presently available computer-aided design systems greatly facilitate the design process, the current practice still requires highly skilled VLSI design engineers to create the necessary structural level hardware descriptions.

There is only a small number of VLSI designers who possess the highly specialized skills needed to create structural level integrated circuit hardware descriptions. Even with the assistance of available VLSI CAD tools, the design process is time consuming and the probability of error is also high because of human involvements. There is a very significant need for a better and more cost effective way to design custom integrated circuits.

SUMMARY OF THE INVENTION

In accordance with the present invention a CAD (computer-aided design) system and method is provided which enables a user to define the functional requirements for a desired target integrated circuit, using an easily understood functional architecture independent level representation, and which generates therefrom the detailed information needed for directly producing an application specific integrated circuit (ASIC) to carry out those specific functions. Thus, the present invention, for the first time, opens the possibility for the design and production of ASICs by designers, engineers and technicians who may not possess the specialized expert knowledge of a highly skilled VLSI design engineer.

The functional architecture independent specifications of the desired ASIC can be defined in a suitable manner, such as in list form or preferably in a flowchart format. The flowchart is a highly effective means of describing a sequence of logical operations, and is well understood by software and hardware designers of varying levels of expertise and training. From the flowchart (or other functional specifications), the system and method of the present invention translates the functional architecture independent specifications into structural an architecture specific level definition of an integrated circuit, which can be used directly to produce the ASIC. The structural level definition includes a list of the integrated circuit hardware cells needed to achieve the functional specifications. These cells are selected from a cell library of previously designed hardware cells of various functions and technical specifications. The system also generates data paths among the selected hardware cells. In addition, the present invention generates a system controller and control paths for the selected integrated circuit hardware cells. The list of hardware cells and their interconnection requirements may be represented in the form of a netlist. From the netlist it is possible using either known manual techniques or existing VLSI CAD layout systems to generate the detailed chip level geometrical information (e.g. mask data) required to produce the particular application specific integrated circuit in chip form.

The preferred embodiment of the system and method of the present invention which is described more fully hereinafter is referred to as a Knowledge Based Silicon Compiler (KBSC). The KBSC is an ASIC design methodology based upon artificial intelligence and expert systems technology. The user interface of KBSC is a flowchart editor which allows the designer to represent VLSI systems in the form of a flowchart. The KBSC utilizes a knowledge based expert system, with a knowledge base extracted from expert ASIC designers with a high level of expertise in VLSI design to generate from the flowchart a netlist which describes the selected hardware cells and their interconnection requirements.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will be better understood by reference to the detailed description which follows, taken in connection with the accompanying drawings, in which

FIG. 1*a* illustrates a functional level design representation of a portion of a desired target circuit, shown in the form of a flowchart;

FIG. 1*b* illustrates a structural level design representation of an integrated circuit;

FIG. 1*c* illustrates a design representation of a circuit at a physical layout level, such as would be utilized in the fabrication of an integrated circuit chip;

FIG. 2 is a block schematic diagram showing how integrated circuit mask data is created from flowchart descriptions by the KBSC system of the present invention;

FIG. 3 is a somewhat more detailed schematic illustration showing the primary components of the KBSC system;

FIG. 4 is a schematic illustration showing how the ASIC design system of the present invention draws upon selected predefined integrated circuit hardware cells from a cell library;

FIG. 5 is an example flowchart defining a sequence of functional operations to be performed by an integrated circuit;

FIG. 6 is a structural representation showing the hardware blocks and interconnection requirements for the integrated circuit defined in FIG. 5;

FIG. 7 is an illustration of the flowchart editor window;

FIG. 8 is an illustration of the flowchart simulator window;

FIG. 9 is an illustration of the steps involved in cell list generation;

FIG. 10 is an example flowchart for a vending machine system;

FIG. 11 illustrates the hardware components which correspond to each of the three macros used in the flowchart of FIG. 10;

FIG. 12 is an initial block diagram showing the hardware components for an integrated circuit as defined in the flowchart of FIG. 10;

FIG. 13 is a block diagram corresponding to FIG. 12 showing the interconnections between blocks;

FIG. 14 is a block diagram corresponding to FIG. 13 after register optimization; and

FIG. 15 is a block diagram corresponding to FIG. 14 after further optimization.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

FIGS. 1*a*, 1*b* and 1*c* illustrate three different levels of representing the design of an integrated circuit. FIG. 1*a* shows a functional (or behavioral) representation architecture independent in the form of a flowchart. A flowchart is a graphic representation of an algorithm and consists of two kinds of blocks or states, namely actions and conditions (decisions). Actions are conventionally represented in the flowchart by a rectangle or box, and conditions are represented by a diamond. Transitions between actions and conditions are represented by lines with arrows. FIG. 1*b* illustrates a structural (or logic) level representation of an integrated circuit. In this representation, blocks are used to represent integrated architecture specific circuit hardware components for performing various functions, and the lines interconnecting the blocks represent paths for the flow of data or control signals between the blocks. The blocks may, for example, represent hardware components such as adders, comparators, registers, system controllers, etc. FIG. 1*c* illustrates a physical layout level representation of an integrated circuit design, which provides the detailed mask data necessary to actually manufacture the devices and conductors which together comprise integrated circuit.

As noted earlier, the design of an integrated circuit at the structural level requires a design engineer with highly specialized skills and expertise in VLSI design. In the KBSC system of the present invention, however, integrated circuits can be designed at a functional level because the expertise in VLSI design is provided and applied by the invention. Allowing the designer to work with flowcharts instead of logic circuit schematics simplifies the task of designing custom integrated circuits, making it quicker, less expensive and more reliable. The designer deals with an algorithm using simple flowcharts at an architecture independent functional (behavioral) level, and needs to know only the necessary logical steps to complete a task, rather than the specific means for accomplishing the task. Designing with flowcharts requires less work in testing because flowcharts allow the designer to work much closer to the algorithm. On the other hand, previously existing VLSI design tools require the designer to represent an algorithm with complex circuit schematics at a structural level, therefore requiring more work in testing. Circuit schematics make it harder for the designer to cope with the algorithm function which needs to be incorporated into the target design because they intermix the hardware and functional considerations. Using flowcharts to design custom integrated circuits will allow a large number of system designers to access VLSI technology, where previously only a small number of designers had the knowledge and skills to create the necessary structural level hardware descriptions.

The overall system flow is illustrated in FIG. 2. The user enters the functional specifications of the circuit into the knowledge based silicon compiler (KBSC) 10 in the form of a flowchart 11. The KBSC 10 then generates a netlist 15 from the flowchart. The netlist 15 includes a custom generated system controller, all other hardware cells required to implement the necessary operations, and interconnection information for connecting the hardware cells and the system controller. The netlist can be used as input to any existing VLSI layout and routing tool 16 to create mask data 18 for geometrical layout.

System Overview

The primary elements or modules which comprise the KBSC system are shown in FIG. 3. In the embodiment illustrated and described herein, these elements or modules are in the form of software programs, although persons skilled in the appropriate art will recognize that these elements can easily be embodied in other forms, such as in hardware.

Referring more particularly to FIG. 3, it will be seen that the KBSC system 10 includes a program 20 called EDSIM, which comprises a flowchart editor 21 for creating and editing flowcharts and a flowchart simulator 22 for simulation and verification of flowcharts. Actions to be performed by each of the rectangles represented in the flowchart are selected from a macro library 23. A program 30 called PSCS (path synthesizer and cell selector) includes a data and control path synthesizer module 31, which is a knowledge based system for data and control path synthesis. PSCS also includes a cell selector 32 for selecting the cells required for system design. The cell selector 32 selects from a cell

library 34 of previously designed hardware cells the appropriate cell or cells required to perform each action and condition represented in the flowchart. A controller generator 33 generates a custom designed system controller for controlling the operations of the other hardware cells. The knowledge base 35 contains ASIC design expert knowledge required for data path synthesis and cell selection. Thus, with a functional flowchart input, PSCS generates a system controller, selects all other hardware cells, generates data and control paths, and generates a netlist describing all of this design information.

The KBSC system employs a hierarchal cell selection ASIC design approach, as is illustrated in FIG. 4. Rather than generating every required hardware cell from scratch, the system draws upon a cell library 34 of previously designed, tested and proven hardware cells of various types and of various functional capabilities with a given type. The macro library 23 contains a set of macros defining various actions which can be specified in the flowchart. For each macro function in the macro library 23 there may be several hardware cells in the cell library 34 of differing geometry and characteristics capable of performing the specified function. Using a rule based expert system with a knowledge base 35 extracted from expert ASIC designers, the KBSC system selects from the cell library 34 the optimum cell for carrying out the desired function.

Referring again to FIG. 3, the cells selected by the cell selector 32, the controller information generated by the controller generator 33 and the data and control paths generated by the data/control path synthesizer 31 are all utilized by the PSCS program 30 to generate the netlist 15. The netlist is a list which identifies each block in the circuit and the interconnections between the respective inputs and outputs of each block. The netlist provides all the necessary information required to produce the integrated circuit. Computer-aided design systems for cell placement and routing are commercially available which will receive netlist data as input and will lay out the respective cells in the chip, generate the necessary routing, and produce mask data which can be directly used by a chip foundry in the fabrication of integrated circuits.

System Requirements

The KBSC system can be operated on a suitable programed general purpose digital computer. By way of example, one embodiment of the system is operated in a work station environment such as Sun3 and VAXStation-II/GPX Running UNIX Operating System and X Window Manager. The work station requires a minimum of 8 megabytes of main storage and 20 megabytes of hard disk space. The monitor used is a color screen with 8-bit planes. The software uses C programming language and INGRES relational data base.

The human interface is mainly done by the use of pop up menus, buttons, and a special purpose command language. The permanent data of the integrated circuit design are stored in the data base for easy retrieval and update. Main memory stores the next data temporarily, executable code, design data (flowchart, logic, etc.), data base (cell library), and knowledge base. The CPU performs the main tasks of creating and simulating flowcharts and the automatic synthesis of the design.

Flowchart Example

To describe the mapping of a flowchart to a netlist, consider an example flowchart shown in FIG. 5, which is of part of a larger overall system. In this illustrative flowchart, two variables, VAL1 and VAL2 are compared and if they are equal, they are added together. In this instance, the first action (Action 1) involves moving the value of variable VALI to register A. The second action comprises moving the value of variable VAL2 to register B. Condition 1 comprises comparing the values in registers A and B. Action 3 comprises adding the values of registers A and B and storing the result in register C.

In producing an integrated circuit to carry out the function defined in FIG. 5, the KBSC maps the flowchart description of the behavior of the system to interconnection requirements between hardware cells. The hardware cells are controlled by a system controller which generates all control signals. There are two types of variables involved in a system controller:

(1) Input variables: These are generated by hardware cells, and/or are external input to the controller. These correspond to conditions in the flowchart.

(2) Output variables: These are generated by the system controller and correspond to actions in the flowchart.

FIG. 6 illustrates the results of mapping the flowchart of FIG. 5 onto hardware cells. The actions and the conditions in the flowchart are used for cell selection and data and control path synthesis. The VALI register and VAL2 register and the data paths leading therefrom have already been allocated in actions occurring before Action 1 in our example. Action 1 causes generation of the data register A. Similarly, Action 2 causes the allocation of data register B. The comparator is allocated as a result of the comparison operation in Condition 1. The comparison operation is accomplished by (1) selecting a comparator cell, (2) mapping the inputs of the comparator cell to registers A and B, (3) generating data paths to connect the comparator with the registers A and B and (4) generating input variables corresponding to equal to, greater than, and less than for the system controller. Similarly the add operation in Action 3 causes selection of the adder cell, mapping of the adder parameters to the registers and creating the data paths.

Following this methodology, a block list can be generated for a given flowchart. This block list consists of a system controller and as many other blocks as may be required for performing the necessary operations. The blocks are connected with data paths, and the blocks are controlled by the system controller through control paths. These blocks can be mapped to the cells selected from a cell library to produce a cell list.

Interactive Flowchart Editor and Simulator

The creation and verification of the flowchart is the first step in the VLSI design methodology. The translation from an algorithm to an equivalent flowchart is performed with the Flowchart Editor 21 (FIG. 3). The verification of the edited flowchart is performed by the Flowchart Simulator 22 The Flowchart Editor and Simulator are integrated into one working environment for interactive flowchart editing, with a designer friendly interface.

EDSIM is the program which contains the Flowchart Editor 21 and the Flowchart Simulator 22. It also provides functions such as loading and saving flow-

charts. EDSIM will generate an intermediate file, called a statelist, for each flowchart. This file is then used by the PSCS program 30 to generate a netlist.

Flowchart Editor

The Flowchart Editor 21 is a software module used for displaying, creating, and editing the flowchart. This module is controlled through the flowchart editing window illustrated in FIG. 7. Along with editing functions the Flowchart Editor also provides checking of design errors.

The following is a description of the operations of the Flowchart Editor. The main editing functions include, create, edit, and delete states, conditions, and transitions. The create operation allows the designer to add a new state, condition, or transitions to a flowchart. Edit allows the designer to change the position of a state, condition or transition, and delete allows the designer to remove a state, condition or transition from the current flowchart. States which contain actions are represented by boxes, conditions are represented by diamonds, and transitions are represented by lines with arrows showing the direction of the transition.

Edit actions allows the designer to assign actions to each box. These actions are made up of macro names and arguments. An example of arguments is the setting and clearing of external signals. A list of basic macros available in the macro library 23 is shown in Table 1.

TABLE 1

Macro	Description
ADD (A,B,C)	C = A + B
SUB (A,B,C)	C = A − B
MULT (A,B,C)	C = A * B
DIV (A,B,C)	C = A div B
DECR (A)	A = A − 1
INCR (A)	A = A + 1
CLR (A)	A = 0
REG (A,B)	B = A
CMP (A,B)	Compare A to B and set EQ,LT,GT signals
CMP0 (A)	Compare A to 0 and set EQ,LT,GT signals
NEGATE (A)	A = NOT (A)
MOD (A,B,C)	C = A Modulus B
POW (A,B,C)	C = A B
DC2 (A,S1,S2,S3,S4)	Decode A into S1,S2,S3,S4
EC2 (S1,S2,S3,S4,A)	Encode S1,S2,S3,S4 into A
MOVE (A,B)	B = A
CALL sub-flowchart (A,B, . . .)	Call a sub-flowchart. Pass A,B . . .
START (A,B, . . .)	Beginning state of a sub-flowchart
STOP (A,B . . .)	Ending state of a sub-flowchart

The Flowchart Editor also provides a graphical display of the flowchart as the Flowchart Simulator simulates the flowchart. This graphical display consists of boxes, diamonds, and lines as shown in FIG. 7. All are drawn on the screen and look like a traditional flowchart. By displaying the flowchart on the screen during simulation it allows the designer to design and verify the flowchart at the same time.

Flowchart Simulator

The Flowchart Simulator 22 is a software module used for simulating flowcharts. This module is controlled through the simulator window illustrated in FIG. 8. The Flowchart Simulator simulates the transitions between states and conditions in a flowchart. The following is a list of the operations of the Flowchart Simulator:

edit data—Change the value of a register or memory.
set state—Set the next state to be simulated.
set detail or summary display—Display summary or detail information during simulation.
set breaks—Set a breakpoint.
clear breaks—Clear all breakpoints.
show breaks—Display current breakpoints.
step—Step through one transition.
execute—Execute the flowchart.
stop—Stop executing of the flowchart. history ON or history OFF—Set history recording on or off.
cancel—Cancel current operation.
help—Display help screen.
close—Close the simulator window.

The results of the simulation are displayed within the simulator window. Also the editor window will track the flowchart as it is being simulated. This tracking of the flowchart makes it easy to edit the flowchart when an error is found.

Cell Selection

The Cell Selector 32 is a knowledge based system for selecting a set of optimum cells from the cell library 34 to implement a VLSI system. The selection is based on functional descriptions in the flowchart, as specified by the macros assigned to each action represented in the flowchart. The cells selected for implementing a VLSI system depend on factors such as cell function, fabrication technology used, power limitations, time delays etc. The cell selector uses a knowledge base extracted from VLSI design experts to make the cell selection.

To design a VLSI system from a flowchart description of a user application, it is necessary to match the functions in a flowchart with cells from a cell library. This mapping needs the use of artificial intelligence techniques because the cell selection process is complicated and is done on the basis of a number of design parameters and constraints. The concept used for cell selection is analogous to that used in software compilation. In software compilation a number of subroutines are linked from libraries. In the design of VLSI systems, a functional macro can be mapped to library cell.

FIG. 4 illustrates the concept of hierarchical cell selection. The cell selection process is performed in two steps:

(1) selection of functional macros
(2) selection of geometrical cells

A set of basic macros is shown in Table 1. A macro corresponds to an action in the flowchart. As an example, consider the operation of adding A and B and storing the result in C. This function is mapped to the addition macro ADD(X, Y, Z). The flowchart editor and flowchart simulator are used to draw the rectangles, diamonds and lines of the flowchart, to assign a macro selected from the macro library 23 to each action represented in the flowchart, and to verify the functions in flowcharts. The flowchart is converted into an intermediate form (statelist) and input to the Cell Selector.

The Cell Selector uses a rule based expert system to select the appropriate cell or cells to perform each action. If the cell library has a number of cells with different geometries for performing the operation specified by the macro, then an appropriate cell can be selected on the basis of factors such as cell function, process technology used, time delay, power consumption, etc.

The knowledge base of Cell Selector 32 contains information (rules) relating to:

(1) selection of macros
(2) merging two macros

(3) mapping of macros to cells
(4) merging two cells
(5) error diagnostics

The above information is stored in the knowledge base 35 as rules.

Cell List Generation

FIG. 9 shows the cell list generation steps. The first step of cell list generation is the transformation of the flowchart description into a structure that can be used by the Cell Selector. This structure is called the statelist. The blocklist is generated from the statelist by the inference engine. The blocklist contains a list of the functional blocks to be used in the integrated circuit. Rules of the following type are applied during this stage.

map arguments to data paths
map actions to macros
connect these blocks

Rules also provide for optimization and error diagnostics at this level.

The cell selector maps the blocks to cells selected from the cell library 34. It selects an optimum cell for a block. This involves the formulation of rules for selecting appropriate cells from the cell library. Four types of information are stored for each cell. These are:

(1) functional level information: description of the cell at the register transfer level.
(2) logic level information: description in terms of flip-flops and gates.
(3) circuit level information: description at the transistor level.
(4) Layout level information: geometrical mask level specification.

The attributes of a cell are:

cell name
description
function
width
height
status
technology
minimum delay
typical delay
maximum delay
power
file
designer
date
comment
inspector

In the cell selection process, the above information can be used. Some parameters that can be used to map macros to cells are:

(1) name of macro
(2) function to be performed
(3) complexity of the chip
(4) fabrication technology
(5) delay time allowed
(6) power consumption
(7) bit size of macro data paths

Netlist Generation

The netlist is generated after the cells have been selected by PSCS. PSCS also uses the macro definitions for connecting the cell terminals to other cells. PSCS uses the state-to-state transition information from an intermediate form representation of a flowchart (i.e. the statelist) to generate a netlist. PSCS contains the following knowledge for netlist generation:

(1) Data path synthesis
(2) Data path optimization
(3) Macro definitions
(4) Cell library
(5) Error detection and correction

The above information is stored in the knowledge base 35 as rules. Knowledge engineers help in the formulation of these rules from ASIC design experts. The macro library 23 and the cell library 34 are stored in a database of KBSC.

A number of operations are performed by PSCS. The following is a top level description of PSCS operations:

(1) Read the flowchart intermediate file and build a statelist.
(2) current_context = START
(3) Start the inference engine and load the current context rules.
(4) Perform one of the following operations depending upon current_context:
(a) Modify the statelist for correct implementation.
(b) Create blocklist, macrolist and data paths.
(c) Optimize blocklist and datapath list and perform error checks.
(d) Convert blocks to cells.
(e) Optimize cell list and perform error checks.
(f) Generate netlist.
(g) Optimize netlist and perform error checks and upon completion Goto 7.
(5) If current_context has changed, load new context rules.
(6) Goto 4.
(7) Output netlist file and stf files and Stop.

In the following sections, operations mentioned in step 4 are described. The Rule Language and PSCS display are also described.

Rule Language

The rule language of PSCS is designed to be declarative and to facilitate rule editing. In order to make the expert understand the structure of the knowledge base, the rule language provides means for knowledge representation. This will enable the format of data structures to be stated in the rule base, which will enable the expert to refer to them and understand the various structures used by the system. For example, the expert can analyze the structure of wire and determine its components. The expert can then refer these components into rules. If a new object has to be defined, then the expert can declare a new structure and modify some existing structure to link to this new structure. In this way, the growth of the data structures can be visualized better by the expert. This in turn helps the designer to update and append rules.

The following features are included in the rule language:

(i) Knowledge representation in the form of a record structure.
(ii) Conditional expressions in the antecedent of a rule.
(iii) Facility to create and destroy structure in rule actions.
(iv) The assignment statement in the action of a rule.
(v) Facility for input and output in rule actions.
(vi) Provide facility to invoke C functions from rule actions.

The rule format to be used is as follows:

The rule format to be used is as follows:

```
Rule    <number> <context>
If {
        <if-clause>
}
Then    {
        <then-clause>
}
where   <number>        rule number
        <context>       context in which this rule is
                        active
        <if-clause>     the condition part of the rule
        <then-clause>   the action part of the rule
```

Inference Strategy

The inference strategy is based on a fast pattern matching algorithm. The rules are stored in a network and the requirement to iterate through the rules is avoided. This speeds up the execution. The conflict resolution strategy to be used is based on the following:

(1) The rule containing the most recent data is selected.

(2) The rule which has the most complex condition is selected.

(3) The rule declared first is selected.

Rule Editor

PSCS provides an interactive rule editor which enables the expert to update the rule set. The rules are stored in a database so that editing capabilities of the database package can be used for rule editing. To perform this operation the expert needs to be familiar with the various knowledge structures and the inferencing process. If this is not possible, then the help of a knowledge engineer is needed.

PSCS provides a menu from which various options can be set. Mechanisms are provided for setting various debugging flags and display options, and for the overall control of PSCS.

Facility is provided to save and display the blocklist created by the user. The blocklist configuration created by the user can be saved in a file and later be printed with a plotter. Also the PSCS display can be reset to restart the display process.

PSCS Example Rules:

```
Rule 1
     IF     no blocks exist
     THEN   generate a system controller.
Rule 2
     IF     a state exists which has a macro AND
            this macro has not been mapped to a block
     THEN   find a corresponding macro in the library
            and generate a block for this macro.
Rule 3
     IF     there is a transition between two
            states AND there are macros in these
            states using the same argument
     THEN   make a connection from a register
            corresponding to the first macro to
            another register corresponding to the
            second macro.
Rule 4
     IF     a register has only a single connection
            from another register
     THEN   combine these registers into
            a single register.
Rule 5
     IF     there are two comparators AND
            input data widths are of the same size AND
            one input of these is same AND
            the outputs of the comparators are used to
            perform the same operation.
     THEN   combine these comparators into
            a single comparator.
Rule 6
     IF     there is a data without a register
     THEN   allocate a register for this data.
Rule 7
     IF     all the blocks have been interconnected AND
            a block has a few terminals not connected
     THEN   remove the block and its terminals, or
            issue an error message.
Rule 8
     IF     memory is to be used, but a block has not
            been created for it
     THEN   create a memory block with data,
            address, read and write data and
            control terminals.
Rule 9
     IF     a register has a single connection to
            a counter
     THEN   combine the register and the counter;
            remove the register and its terminals.
Rule 10
     IF     there are connections to a terminal of
            a block from many different blocks
     THEN   insert a multiplexor;
            remove the connections to the terminals and
            connect them to the input of the multiplexor;
            connect the output of the
            multiplexor to the input of the block.
```

Additional rules address the following points:

remove cell(s) that can be replaced by using the outputs of other cell(s)

reduce multiplexor trees

use fan-out from the cells, etc.

Soft Drink Vending Machine Controller Design Example

The following example illustrates how the previously described features of the present invention are employed in the design of an application specific integrated circuit (ASIC). In this illustrative example the ASIC is designed for use as a vending machine controller. The vending machine controller receives a signal *each time a coin has been deposited in a coin receiver.* The coin value is recorded and when coins totalling the correct amount are received, the controller generates a signal to dispense a soft drink. When coins totalling more than the cost of the soft drink are received, the controller dispenses change in the correct amount.

This vending machine controller example is patterned after a textbook example used in teaching digital system controller design. See Fletcher, William I., *An Engineering Approach to Digital Design,* Prentice-Hall, Inc., pp. 491–505. Reference may be made to this textbook example for a more complete explanation of this vending machine controller requirements, and for an understanding and appreciation of the complex design procedures prior to the present invention for designing the hardware components for a controller.

FIG. 10 illustrates a flowchart for the vending machine controller system. This flowchart would be entered into the KBSC system by the user through the flowchart editor. Briefly reviewing the flowchart, the controller receives a coin present signal when a coin is received in the coin receiver. State0 and cond0 define a waiting state awaiting deposit of a coin. The symbol CP represents "coin present" and the symbol !CP repre-

sents "coin not present". Statel and condI determine when the coin has cleared the coin receiver. At state20, after receipt of a coin, the macro instruction ADD3.1 (lc, cv, sum) instructs the system to add lc (last coin) and cv (coin value) and store the result as sum. The macro instruction associated with state21 moves the value in the register sum to cv. The macro CMP.1 at state22 compares the value of cv with PR (price of soft drink) and returns signals EQ, GT and LT. The condition cond2 tests the result of the compare operation CMP.1. If the result is "not greater than" (!GT.CMP.1), then the condition cond3 tests to see whether the result is "equal" (EQ.CMP.1). If the result is "not equal" (!EQ.CMP.1), then control is returned to state0 awaiting the deposit of another coin. If cond3 is EQ, then state4 generates a control signal to dispense a soft drink (droppop) and the macro instruction CLR.1(cv) resets cv to zero awaiting another customer.

If the total coins deposited exceed the price, then state30 produces the action "returncoin". Additionally, the macro DECR.1 (cv) reduces the value of cv by the amount of the returned coin. At state31 cv and PR are again compared. If cv is still greater than PR, then control passes to state30 for return of another coin. The condition cond5 tests whether the result of CMP.2 is EQ and will result in either dispensing a drink (droppop) true or branching to state0 awaiting deposit of another coin. The macros associated with the states shown in FIG. 10 correspond to those defined in Table 1 above and define the particular actions which are to be performed at the respective states.

Appendix A shows the intermediate file or "statelist" produced from the flowchart of FIG. 10. This statelist is produced as output from the EDSIM program 20 and is used as input to the PSCS program 30 (FIG. 3).

FIG. 11 illustrates for each of the macros used in the flowchart of FIG. 10, the corresponding hardware blocks. It will be seen that the comparison macro CMP (A,B) results in the generation of a register for storing value A, a register for storing value B, and a comparator block and also produces control paths to the system controller for the EQ, LT, and GT signals generated as a result of the comparison operation. The addition macro ADD (A,B,C) results in the generation of a register for each of the input values A and B, a register for the output value C, and in the generation of an adder block. The macro DECR (A) results in the generation of a counter block. The PSCS program 30 maps each of the macros used in the flowchart of FIG. 10 to the corresponding hardware components results in the generation of the hardware blocks shown in FIG. 12. In generating the illustrated blocks, the PSCS program 30 relied upon rules 1 and 2 of the above listed example rules.

FIG. 13 illustrates the interconnection of the block of FIG. 12 with data paths and control paths. Rule 3 was used by the data/control path synthesizer program 31 in mapping the data and control paths.

FIG. 14 shows the result of optimizing the circuit by applying rule 4 to eliminate redundant registers. As a result of application of this rule, the registers R2, R3, R7, R8, and R9 in FIG. 13 were removed. FIG. 15 shows the block diagram after further optimization in which redundant comparators are consolidated. This optimization is achieved in the PSCS program 30 by application of rule 5.

Having now defined the system controller block, the other necessary hardware blocks and the data and control paths for the integrated circuit, the PSCS program 30 now generates a netlist 15 defining these hardware components and their interconnection requirements. From this netlist the mask data for producing the integrated circuit can be directly produced using available VLSI CAD tools.

```
name rpop;
data path @ic<0:5>, cv<0:5>, sum<0:5>, @pr<0:5>;
{
state4 : state0;
state30 : state31;
state21 : state22;
state20 : state21;
state0 :. kp state0;
state0 :. ep state1;
state1 :. ep state1;
state1 :. kp state20;
state22 :. GT.CMP.1 state30;
state22 :. !GT.CMP.1*EQ.CMP.1 state4;
state22 :. !GT.CMP.1*!EQ.CMP.1 state0;
state31 :. GT.CMP.2 state30;
state31 :. !GT.CMP.2*EQ.CMP.2 state4;
state31 :. !GT.CMP.2*!EQ.CMP.2 state0;
state30 = returncoin;
state30 = DECR.1(cv);
state4 = droppop;
state4 = CLR.1(cv);
state31 = CMP.2(cv,pr);
state22 = CMP.1(cv,pr);
state21 = MOVE.1(sum,cv);
state20 = ADD3.1(ic,cv,sum);
}
```

That which I claimed is:

1. A computer-aided design system for designing an application specific integrated circuit directly from architecture independent functional specifications for the integrated circuit, comprising

a macro library defining a set of architecture independent operations comprised of actions and conditions;

input specification means operable by a user for defining architecture independent functional specifications for the integrated circuit, said functional specifications being comprised of a series of operations comprised of actions and conditions, said input specification means including means to permit the user to specify for each operation a macro selected from said macro library;

a cell library defining a set of available integrated circuit hardware cells for performing the available operations defined in said macro library;

cell selection means for selecting from said cell library for each macro specified by said input specification means, appropriate hardware cells for performing the operation defined by the specified macro, said cell selection means comprising an expert system including a knowledge base containing rules for selecting hardware cells from said cell library and inference engine means for selecting appropriate hardware cells from said cell library in accordance with the rules of said knowledge base; and

netlist generator means cooperating with said cell selection means for generating as output from the system a netlist defining the hardware cells which are needed to achieve the functional requirements of the integrated circuit and the connections therebetween.

2. The system as defined in claim 1 wherein said input means comprises means specification for receiving user

input of a list defining the series of actions and conditions.

3. The system as defined in claim 1 additionally including mask data generator means for generating from said netlist the mask data required to produce an integrated circuit having the specified functional requirements.

4. The system as defined in claim 1 wherein said input means comprises flowchart editor means specification for creating a flowchart having elements representing said series of actions and conditions.

5. The system as defined in claim 4 additionally including flowchart simulator means for simulating the functions defined in the flowchart to enable the user to verify the operation of the integrated circuit.

6. The system as defined in claim 1 additionally including data path generator means cooperating with said cell selection means for generating data paths for the hardware cells selected by said cell selection means.

7. The system as defined in claim 6 wherein said data path generator means comprises a knowledge base containing rules for selecting data paths between hardware cells and inference engine means for selecting data paths between the hardware cells selected by said cell selection means in accordance with the rules of said knowledge base and the arguments of the specified macros.

8. The system as defined in claim 6 additionally including control generator means for generating a controller and control paths for the hardware cells selected by said cell selection means.

9. A computer-aided design system for designing an application specific integrated circuit directly from a flowchart defining architecture independent functional requirements of the integrated circuit comprising

a marco library defining a set of architecture independent operations comprised of actions and conditions;

flowchart editor means operable by a user for creating a flowchart having elements representing said architecture independent operations;

said flowchart editor means including macro specification means for permitting the user to specify for each operation represented in the flowchart a macro selected from said macro library;

a cell library defining a set of available integrated circuit hardware cells for performing the available operations defined in said macro library;

cell selection means for selecting form said cell library for each specified macro, appropriate hardware cells for performing the operation defined by the specified macro, said cell selection means comprising an expert system including a knowledge base containing rules for selecting hardware cells from said cell library and inference engine means for selecting appropriate hardware cells from said cell library in accordance with the rules of said knowledge base; and

data path generator means cooperating with said cell selection means for generating data paths for the hardware cells selected by said cell selector means, said data path generator means comprising a knowledge base containing rules for selecting data paths between hardware cells and inference engine means for selecting data paths between hardware cells selected by said cell selection means in accordance with the rules of said knowledge base and the arguments of the specified macros.

10. The system as defined in claim 9 additionally including control generator means for generating a controller and control paths for the hardware cells selected by said cell selection means.

11. A computer-aided design system for designing an application specific integrated circuit directly from a flowchart defining architecture independent functional requirements of the integrated circuit, comprising

flowchart editor means operable by a user for creating a flowchart having boxes representing architecture independent actions, diamonds representing architecture independent conditions, and lines with arrows representing transitions between actions and condition and including means for specifying for each box or diamond, a particular action or condition to be performed;

a cell library defining a set of available integrated circuit hardware cells for performing actions and conditions;

a knowledge base containing rules for selecting hardware cells from said cell library and for generating data and control paths for hardware cells; and

expert system means operable with said knowledge base for translating the flowchart defined by said flowchart editor means into a netlist defining the necessary hardware cells and data and control paths required in an integrated circuit having the specified functional requirements.

12. The system as defined in claim 11 including mask data generator means for generating from said netlist the mask data required to produce an integrated circuit having the specified functional requirements.

13. A computer-aided design process for designing an application specific integrated circuit which will perform a desired function comprising

storing a set of definitions of architecture independent actions and conditions;

storing data describing a set of available integrated circuit hardware cells for performing the actions and conditions defined in the stored set;

storing in an expert system knowledge base a set of rules for selecting hardware cells to perform the actions and conditions;

describing for a proposed application specific integrated circuit a series of architecture independent actions and conditions;

specifying for each described action and condition of the series one of said stored definitions which corresponds to the desired action or condition to be performed; and

selecting from said stored data for each of the specified definitions a corresponding integrated circuit hardware cell for performing the desired function of the application specific integrated circuit, said step of selecting a hardware cell comprising applying to the specified definition of the action or condition to be performed, a set of cell selection rules stored in said expert system knowledge base and generating for the selected integrated circuit hardware cells, a netlist defining the hardware cells which are needed to perform the desired function of the integrated circuit and the interconnection requirements therefor.

14. A process as defined in claim 13, including generating from the netlist the mask data required to produce an integrated circuit having the desired function.

15. A process as defined in claim 13 including the further step of generating data paths for the selected integrated circuit hardware cells.

16. A process as defined in claim 15 wherein said step of generating data paths comprises applying to the selected cells a set of data path rules stored in a knowledge base and generating the data paths therefrom.

17. A process as defined in claim 16 including the further step of generating control paths for the selected integrated circuit hardware cells.

18. A knowledge based design process for designing an application specific integrated circuit which will perform a desired function comprising

storing in a macro library a set of macros defining architecture independent actions and conditions;

storing in a cell library a set of available integrated circuit hardware cells for performing the actions and conditions;

storing in a knowledge base set of rules for selecting hardware cells from said cell library to perform the actions and conditions defined by the stored macros;

describing for a proposed application specific integrated circuit a flowchart comprised of elements representing a series of architecture independent actions and conditions which carry out the function to be performed by the integrated circuit;

specifying for each described action and condition of said series a macro selected from the macro library which corresponds to the action or condition; and

applying rules of said knowledge base to the specified macros to select from said cell library the hardware cells required for performing the desired function of the application specific integrated circuit and generating for the selected integrated circuit hardware cells, a netlist defining the hardware cells which are needed to perform the desired function of the integrated circuit and the interconnection requirements therefor.

19. A process as defined in claim 18 also including the steps of

storing in said knowledge base a set of rules for creating data paths between hardware cells, and

applying rules of said knowledge base to the specified means to create data paths for the selected hardware cells.

20. A process as defined in claim 19 also including the steps of generating a controller and generating control paths for the selected hardware cells.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 4,922,432

DATED : May 1, 1990

INVENTOR(S) : Hideaki Kobayashi, et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

ON TITLE PAGE: under the section "References Cited" under "Other Publications":

"Verifying Compiled Silicon", by E. K. cheng, VLSI Design, Oct. 1984, pp. 1-4." should be -- "Verifying Compiled Silicon", by E. K. Cheng, VLSI Design, Oct. 1984, pp. 1-4." --.

"quality of Designs from An Automatic Logic Generator", by T. D. Friedman et al., IEEE 7th DA Conference, 1970, pp. 71-89." should be -- "Quality of Designs from An Automatic Logic Generator", by T. D. Friedman et al., IEEE 7th DA Conference, 1970, pp. 71-89. --.

"Trevillyan-Trickey, H., Flamel: *A High Level Hardward Compiler*, IEEE Transactions On Computer Aided Design, Mar. 1987, pp. 259-269." should be -- Trevillyan-Trickey, H., Flamel: *A High Level Hardware Compiler*, IEEE Transactions On Computer Aided Design, Mar. 1987, pp. 259-269. --.

In the abstract:

Every occurrence of "functional architecture independent" should be -- architecture independent functional --.

Column 1, line 19, "a" should be -- an --.

[redacted]

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 4,922,432

DATED : May 1, 1990

INVENTOR(S) : Hideaki Kobayashi, et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 2, line 10, "functional architecture independent" should be -- architecture independent functional --.

Column 2, line 21, "functional architecture independent" should be -- architecture independent functional --.

Column 2, lines 29-30, "functional architecture independent" should be -- architecture independent functional --.

Column 2, line 31, "structural" should be after "specific".

Column 3, lines 51-52, "representation" should be after "architecture independent".

Column 3, lines 61-62, "integrated" should be after "specific".

Column 6, line 62, after "22" insert -- . --.

Column 7, line 43 (in Table 1), "C = A B" should be -- C = A^B --.

Column 8, line 9 should end with the word "flowchart" and "history" should begin on the next line.

[redacted]

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 4,922,432

DATED : May 1, 1990

INVENTOR(S) : Hideaki Kobayashi, et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 10, line 23, "data paths" should be -- datapaths --.

Column 10, line 68, delete "The rule format to be used is as follows:".

Column 12, line 54, "Enqineering" should be -- Engineering --.

Column 13, line 55, "block" should be -- blocks --.

In the Claims:

Column 14, line 68, before "means" (first occurrence) insert -- specification --; after "means" (second occurrence) delete "specification".

Column 15, line 9, before "means" (first occurrence) insert -- specification --; after "means" (second occurrence) delete "specification".

Column 15, line 35, after "circuit" insert -- , --.

Column 15, line 36, "marco" should be -- macro --.

Column 15, line 49, "form" should be -- from --.

[redacted]

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 4,922,432

DATED : May 1, 1990

INVENTOR(S) : Hideaki Kobayashi, et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 16, line 14, "condition" should be -- conditions --.

Column 17, line 19, after "base" insert -- a --.

Signed and Sealed this

Fourteenth Day of January, 1992

Attest:

Attesting Officer

HARRY F. MANBECK, JR.

Commissioner of Patents and Trademarks

[redacted]

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

RICOH COMPANY, LTD., Plaintiff, vs. AEROFLEX INCORPORATED, AMI SEMICONDUCTOR, INC., MATROX ELECTRONIC SYSTEMS, LTD., MATROX GRAPHICS INC., MATROX INTERNATIONAL CORP., and MATROX TECH, INC., Defendants.	Case No. CV 03-04669 MJJ **[PROPOSED] ORDER GRANTING DEFENDANTS' MOTION FOR JUDGMENT ON THE PLEADINGS PURSUANT TO RULE 12(C)** Date: March 16, 2004 Time: 9:30 a.m. Ctrm: 11, 19th Floor Judge: Hon. Martin J. Jenkins

This matter came before the Court on March 16, 2004 on motion by Defendants Aeroflex, Inc., AMI Semiconductor, Inc., Matrox Electronic Systems, Ltd., Matrox Graphics Inc., Matrox International Corp., and Matrox Tech, Inc. (" Defendants") seeking a judgment from the Court, pursuant to Rule 12(c) of the Federal Rules of Civil Procedure, that they do not infringe claims 13 -20 of United States Patent No. 4,922,432 (the "'432 patent") under 35 U.S.C. § 271(g).

After consideration of the papers filed in support of the motion, any papers filed in opposition, and any oral argument of counsel:

IT IS ORDERED, ADJUDGED AND DECREED that:

1. Defendants' motion is granted.

Dated: _____, 2004

_____

The Honorable Martin J. Jenkins
United States District Court Judge

Submitted February 10, 2004 by:
HOWREY SIMON ARNOLD & WHITE, LLP

By: /s/ Thomas Mavrakakis
Teresa M. Corbin
Attorneys for Defendants
AEROFLEX INCORPORATED, AMI SEMICONDUCTOR, INC., MATROX ELETRONIC SYSTEMS, LTD., MATROX GRAPHICS INC., MATROX INTERNATIONAL CORP., and MATROX TECH, INC.